DISTRIBUTED LINEAR COMBINATION ESTIMATORS FOR LOCALIZATION BASED ON RECEIVED SIGNAL STRENGTH

A Dissertation

by

WEI-YU CHEN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Scott L. Miller
Committee Members,	Costas N. Georghiades
	Jean-François Chamberland
	Alex Sprintson
	Riccardo Bettati
Head of Department,	Chanan Singh

December 2013

Major Subject: Electrical and Computer Engineering

ABSTRACT

Locating the position of a radio frequency device is indispensable in many wireless applications. The most famous method is the Global Positioning System (GPS), which uses trilateration with satellites, is generally unavailable for indoor devices and expensive for large networks. Therefore, this dissertation aims to develop and discuss accurate, fast, low-cost, energy-efficient, and robust localization algorithms especially based on the received signal strength (RSS).

This dissertation proposes a distributed and iterative estimator by linearly combining location estimates from maximum likelihood based range estimates. In non-cooperative cases where unknown-location (blindfolded) devices only utilize the information from known-location devices (anchors), each combining weight is proportional to the reciprocal of the estimated distance squared between the blindfolded node and an anchor. The numerical simulations demonstrate that the proposed LC estimator has similar error behaviors to the maximum likelihood estimator (MLE) and fewer computations under various topologies and noisy wireless environments. If the parameters for the RSS model are unknown, they are estimated by the least square and/or maximum likelihood methods. The accuracy difference of the linear combination estimators by estimated and perfect parameters is acceptable and decreasing as more anchors are deployed.

In cooperative localization, a blindfolded node uses information from not only anchors but also other blindfolded nodes. The combining weight is now proportional to the reciprocal of the estimated distance squared and the transmitter's positioning error. After being mainly compared with the distributed maximum likelihood estimator by coordinate descent method and the distributed weighted-multidimensional

scaling (dwMDS) method, the LC estimator performs well in accuracy, computation time, and the use of wireless transmissions under various topologies, connectivities, and noisy environments. Moreover, the estimation error is clipped by upper and lower bounds. The drawback is that the convergence is not guaranteed, although non-convergent cases rarely happen. For the connectivity issue, placing more nodes with smaller transmitting ranges results in fewer connected nodes and less power consumption. However, to improve localization of an existing system, the relative costs of node and consumed power must be considered to determine the lowest cost system. Finally, the density of blindfolded nodes is two to three times to the density of anchors to achieve the same accuracy.

ACKNOWLEDGEMENTS

Obtaining my Ph.D. degree at Texas A&M could not have been possible without the support of numerous people. Foremost, I would like to express my gratitude to my research advisor, Professor Scott L. Miller. What I have learned from him is not only knowledge and experience but also his attitude and philosophy to do research and to be a good human being. I am also indebted to my phenomenal committee members: Professor Costas N. Georghiades, Professor Jean-François Chamberland, Professor Alex Sprintson, and Professor Riccardo Bettati. In addition, I would like to give thanks to Professor Andrew K. Chan, who was in my committee but retired before my defense, and Professor Radu Stoleru, who substituted for Professor Bettati in my preliminary exam. They all gave me many valuable suggestions and advises for my research and my life.

The Information Science and Systems group in Electrical and Computer Engineering at Texas A&M University is an excellent place for graduate study. Our group has many distinguished professors to provide informative courses. Besides my advisor and committee members, I want to express my gratitude for Professor Krishna Narayanan, Professor Henry Pfister, and Professor Serap Savari for their efforts of teaching. I would like to thank all students in our group, especially Lingjia Liu, Parimal Parag, Anantharaman Balasaubaramanian, Fan Zhang, Meng Zeng, Yung-Yih Jian, and Jerry Huang for their valuable discussions.

I wish to show my appreciation of the department of Electrical and Computer Engineering and Texas A&M University. Both of them provide good environments for students. The advisor Tammy Carda in our department and office associates in our group Gayle Travis and Paula Evans also help me a lot. I want to thank their

services for students.

The Taiwanese student association at Texas A&M is the first station for Taiwanese Aggies. I express my thanks to Professor Je-Chin Han and his wife in Mechanical Engineering, Professor Kuang-An Chang in Civil Engineering, and many members for their help in daily life. The softball team of Taiwanese student association gives me a wonderful environment to relax, where I had unbelievable teammates and games. I would like to thank them and will always remember these moments. Many of my new and old friends around the world have encouraged and helped me in these years. I deeply express my gratitude here.

My family has always been the biggest support in my life and I would like to show my appreciation of them. My parents Ching-Jyh Chen and Yen-Yu Lee always physically and mentally support me and encourage me; my brother Chiang-Yu Chen discusses many research problems and approaches with me and also shares experience in graduate study. My wife Mu-Fen Hsieh helps me take care of our son Caden. Although this two-year-old boy only knows a little about what his father is doing, his smile is a big motivation for me to finish my degree. Therefore, I dedicate this dissertation to my family.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Wireless Localization

The thriving development of wireless devices urgently demands the accuracy of locating the position of a radio frequency (RF) device in applications such as public safety (Emergency 911 [2]), environmental monitoring, traffic control, and inventory management. As the popularity of personal mobile equipments rises with powerful computing ability, many personal applications and commercial services also require positioning. For example, a developing standard for wireless communications: LTE (Long-term Evolution) considers many positioning methods to meet regulatory requirements and the LPP (LTE positioning protocol) can be found in the 36.355 specification [3]. Besides location-based applications, knowing position improves routing in a network [4]. Location-Aided Routing (LAR) [5] reduces the searching space of routing nodes and Location-based Routing (LR) [6] decreases overhead and improves scalability. Even simple localization without any known location makes routing efficient [7].

The general configuration for these applications consists of a wireless network of nodes, which either have known locations (referred to as anchors or beacons) or unknown locations (referred to as blindfolded nodes). The anchors can be in either absolute or relative coordinates. Localization aims to accurately and efficiently determine the positions of blindfolded nodes in the wireless network. Many localization problems can be viewed as inverse problems by utilizing measurements to infer the coordinates of the blindfolded nodes [8]. To compensate for the lack of anchors in a network, Niculescu and Nath proposed a "DV-hop" approach [9], which forms a fixed frame to determine the positions. Another famous method is AFL (Anchor-Free

Localization), which uses the idea of fold-freedom to build a rigid graph [10, 11].

The most famous locating method is the Global Positioning System (GPS), which uses trilateration by arrival time with at least four satellites [12, 13]. The trilateration method provides the estimated location in the intersection of several spheres where the centers are the locations of transmitters and the radii are the measured distances between transmitters and the blindfolded receiver. The GPS was first developed by the U.S. Department of Defense in 1973 and nowadays GPS navigations are affordable and popularly used by civilians. Although the technology of GPS is mature, it requires lines of sight from satellites to a located device which is unavailable for indoor use. Moreover, for a large network such as a wireless sensor network (WSN), equipping every sensor with a GPS is expensive in terms of both device cost and power consumption. These two issues drive researchers to invent new localization methods.

There are many localization methods and several good literature surveys can be found in [14, 15, 16]. The taxonomy of localization is listed as follows.

1. Measurement types:

   (a) Range-free: As the name describes, rang-free (or measurement-free) techniques only use the content of the message and therefore require no ranging hardware [17]. A famous range-free algorithm is an approximate point in triangle (APIT) test by He et al. [18].

   (b) Range-based: Range-based (or measurement-based) methods utilize measurements to estimate unknown locations [14]; common measurements used in localization are:

      i. Time of arrival (TOA): The arrival time of a signal from a transmitter to a receiver increases as their distance increases. Because a TOA

measurement indicates the time when the first signal reaches the receiver, whether a line-of-sight (LOS) exists affects the range accuracy most. The time delay model of TOA is represented as a Gaussian random variable. Because of the additive noise feature, TOA methods have good statistical properties but the extra (hardware) cost is required to maintain precise synchronization. Pioneer research of TOA can be found in [19].

A variation of TOA is called time difference of arrival (TDOA). A TDOA method computes the differences in the arrival times of multiple time measurements and several methods are introduced in [20].

ii. Received signal strength (RSS): The power decays as it travels farther and therefore the strength of the received power indicates the distance between the transmitter and the receiver. RSS can be measured without extra bandwidth and energy which is an advantage. Moreover, the extra hardware cost for the RSS is almost zero because most of wireless devices already have built-in equipments to measure the power strength. The fluctuation of power arises from multipaths and shadowing. Multipath signals caused by frequency-selective fading can be diminished by spread-spectrum or ultra-wide band (UWB) systems [21]. Thus, the main range error of RSS is the shadowing effect caused by the environment and this error is usually larger compared to TOA.

The power loss is then modeled by the log-normal distribution. Thus Gaussian noise in the exponent makes statistical signal processing difficult. This also makes the classical least square method perform poorly in RSS model.

iii. Angle of arrival (AOA): AOA methods provide the relative directions of pairwise nodes with antenna arrays or directional antennas [22]. In antenna arrays, the AOA actually obtains directions by computing different time arrivals. Thus the measurement error is similar to that in TOA and is modeled as a Gaussian distribution.

2. Centralized versus distributed: A centralized method requires a fusion center (central processor) to gather all information in the network and then inform all blindfolded nodes their locations. For example, Doherty et al. proposed a centralized method using semidefinite programming (SDP) [23], and the relaxation of the SDP method was proposed in [24]. MDS-MAP, which uses the estimated distances and multidimensional scaling (MDS) to form a least square problem, is another centralized algorithm [25].

   In contrast, all blindfolded nodes can locate themselves with local information in a distributed algorithm. Because of scalability for large networks such as wireless sensor networks (WSN), distributed localization is more popular such as in the DV approach mentioned in [9]. The distributed weighted-multidimensional scaling (dwMDS), based on MDS in [26], is another example.

3. Non-cooperative versus cooperative: In non-cooperative localization, only anchors communicate with blindfolded nodes [27]. An estimated location of a blindfolded node is computed with only the anchor's information. On the other hand, if blindfolded nodes can help one another position, the method is called cooperative localization. The concept of cooperative localization is first introduced by Savarese et al. [28]. Cooperation presumably provides better estimations if the accuracy of blindfolded nodes is within a reasonable range.

1.2 Problem and Achievement

The goal of localization is to provide accurate and fast positioning with low cost, little communication, and low power consumption in different topologies and wireless environments. This dissertation focuses on the received signal strength (RSS) model to accomplish the above goals.

The existing localization algorithms have certain drawbacks and need improvement. For example, one challenge is to compensate for the unavailability of GPS. Another challenge is the feasibility of the most popular estimation method, maximum likelihood estimation (MLE). There is no closed form solution for MLE in both TOA and RSS localizations because the unknown locations are contained in the estimated ranges. A modification of MLE is approximate maximum likelihood (AML), which is solved by least square method [29]. Although the approximated method works well in the TOA model, its performance is disappointing in the RSS model.

To quantify the accuracy of estimation, the mean square error (MSE) is usually used. The unbiased Cramér-Rao bound (CRB) is the lowest achievable MSE for unbiased estimators and serves as the reference of performance. The CRBs for both RSS and TOA localization problems have been discussed in [30]. However, unbiased estimators are either impossible or have different behaviors from most of real estimators based on RSS. Elnahrawy et al. characterize the fundamental limit of indoor RSS localization [31] and provide more realistic comparison. Power consumption is determined by both computations and communications. An efficient method which quickly computes in local and has less inter-node communication is desired. Thus, all these issues will serve as performance measures to compare different localization methods.

This dissertation provides a novel distributed and iterative estimator based on

a linear combination of multiple location estimates to meet the chanllenge. The proposed algorithm works on both non-cooperative and cooperative modes in indoor as well as outdoor environments. The new method will then be compared with other related methods in terms of accuracy, computation time, and the use of wireless transmissions under different wireless environments and topologies.

This dissertation is organized as follows. In Section 2, the RSS model will first be reviewed and the range estimators will be used in the proposed method. The new linear combination (LC) algorithm is derived in non-cooperative localization. The performance is compared with the maximum likelihood estimator (MLE) in accuracy and efficiency in computer simulation. A lower bound, an upper bound, and an approximation will also be given for accuracy analysis. This section ends by presenting the estimations of channel parameters for RSS and applying the estimated parameters to the LC estimators. Section 3 extends my work to cooperative cases. Besides my algorithm, the MLE by coordinate descent method, dwMDS, sequential greedy optimization (SGO) [32], and distributed spatially constrained localization (DSCL) [33] are compared in many situations. Error bounds and approximation are presented in an iterative manner. The connectivity is discussed at the end of Section 3. Finally, Section 4 draws conclusions and discusses possible future research.

2. NON-COOPERATIVE LOCALIZATION

The current section concerns non-cooperation localization, which considers the situation where blindfolded nodes only receive but do not transmit signals in a wireless network [27]. The term "non-cooperative" means that blindfolded nodes do not help one another to improve their localization. In contrast, cooperative localization, where blindfolded nodes serve as both transmitters and receivers, is discussed in the next section.

Section 2.1 discusses range estimators, which are necessary for range-based localization, including non-cooperative localization and cooperative localization. A mathematical model of non-cooperative localization is provided and the maximum likelihood estimator (MLE) is given based on the model in Section 2.2. A new proposed linear combination (LC) estimator is presented in Section 2.3 and lower and upper bounds on its performance are derived in Section 2.4. Section 2.5 shows the MLE compared with LC in simulations under two basic topologies with various parameters. The complexity for both algorithms and the convergence of the LC method are discussed in Sections 2.6 and 2.7, respectively. In the final section of this section, Section 2.8, the estimation of system parameters which are previously assumed known are investigated. Localization using known and estimated parameters are then compared.

2.1 Range Estimators

Essential for range-based localization algorithms, a range estimator provides the estimated distance between two devices. For example, the MDS method [26] and our proposed linear combination estimator in the following sections both require range estimates.

2.1.1 Time of Arrival (TOA)

The range estimate based on time of arrival (TOA) is obtained by dividing travel time by speed of propagation medium [13, 14]. The TOA model is compared with the received signal strength model, the main focus in the dissertation. The time of arrival is modeled as a Gaussian random variable T:

$$T = \frac{d}{c_p} + Z_T \tag{2.1}$$

where d is the distance between a transmitter and a receiver, Z_T represents an additive zero mean Gaussian noise with variance σ_T, and v_p is the propagation speed of the media which could be microwave or acoustics. For instance, $v_p = 3 \times 10^8$ meter/second for the speed of light.

The time difference of arrival (TDOA) which computes the difference of multiple time measurements to obtain distance is a variation of TOA and TDOA was early introduced in [34]. There are two different types of TDOA [35]: 1) a single signal transmits to multiple receivers at different positions, e.g. an uplink signal from a mobile device to multiple base stations. 2) multiple signals from a single node to another node, e.g., a node equips ultrasound or acoustic as the second propagation method as well as radio frequency for the build-in first signal. Then two media's different propagation speeds are used to computes the distance. It should be noticed that TOA and first type of TDOA require precise time synchronization.

2.1.2 Received signal strength (RSS)

In received signal strength (RSS) based localization, a transmitter sends an RF signal to a receiver, which must then estimate their relative distance based on an RSS measurement. The relationship between range and received power is described

by a typical log-distance model with log-normal shadowing [36, 37]. In particular, given the distance $D = d$ (D is a random variable), the received power, P, is another random variable given by

$$P = p_0 \left(\frac{d}{d_0}\right)^{-n_p} 10^{\frac{Z}{10}}, \tag{2.2}$$

where p_0 is the received power that would be measured at some nominal distance, d_0. In addition, n_p is the propagation loss exponent, and Z is a zero-mean Gaussian random variable with variance σ^2 which represents random shadowing. If there is no further mention, $(n_p, \sigma, p_0(d_0))$ are treated as known model parameters. To avoid near-field effect, the reference distance d_0 must be in the far field of the antenna, i.e., d_0 has to be larger than the far-field distance d_f [36]. A typical $d_0 = 1$ meter as for IEEE 802.15.4 [38]. The value of n_p is 2 in free space. Typical values of n_p in buildings with line of sight range from 1.6 to 1.8. For cellular radio, n_p ranges from 2.7 to 5. For obstructed propagation in buildings and factories, it is 2-3 and 4-6, respectively [36]. The value of σ usually ranges from 5 to 12, depending on frequency, antenna heights, and environments [37, 39]. The value is smaller in open areas and larger in suburbs.

The above power/range relationship can be expressed in decibels as

$$P_{\text{dB}} \equiv \rho + Z = p_{0,\text{dB}} - 10n_p \log_{10}\left(\frac{d}{d_0}\right) + Z. \tag{2.3}$$

Hence, P_{dB} is a Gaussian random variable with mean $\rho = p_{0,\text{dB}} - 10n_p \log_{10}(\frac{d}{d_0})$ and P follows a log-normal distribution. More explicitly,

$$f_{P_{\text{dB}}}(p_{\text{dB}}|D = d) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(p_{\text{dB}} - \rho)^2}{2\sigma^2}\right). \tag{2.4}$$

An RSS range estimator strives to estimate the unknown distance, d, given an ob-

servation of $P = p$ or equivalently $P_{\text{dB}} = p_{\text{dB}}$.

The Cramér-Rao bound (CRB) or Cramér-Rao lower bound (CRLB) provides the lowest achievable mean squared error for an estimator [40]. The CRB can be used as a baseline to compare other algorithms, especially for unbiased estimators. Since (2.4) satisfies the regularity condition where both $\frac{\partial f_{P_{\text{dB}}}(p_{\text{dB}}|D=d)}{\partial d}$ and $\frac{\partial^2 f_{P_{\text{dB}}}(p_{\text{dB}}|D=d)}{\partial d^2}$ are absolutely integrable with respect to p [40], then for any unbiased estimate $\hat{d}$

$$\text{Var}(\hat{d}) \geq \frac{1}{I(d)} = \frac{1}{-\mathbb{E}\left[\frac{\partial^2 \ln f_{P_{\text{dB}}}(p_{\text{dB}}|D=d)}{\partial d^2}\right]} = \left(\frac{\sigma \ln 10}{10 n_p}\right)^2 d^2 \equiv b^2 d^2 \tag{2.5}$$

where $I(d)$ is a Fisher information matrix (FIM) and $b \equiv \frac{\sigma \ln 10}{10 n_p}$. Furthermore, the estimators which can attain the CRB are called *efficient* [41].

The simplest and most common estimator is the maximum likelihood estimator (MLE) [42] that forms its estimate as the value of d which maximizes the conditional PDF in (2.4). The maximum likelihood (ML) range estimator can then be obtained as

$$\hat{D}_{\text{ML}}(P) = d_0 \left(\frac{P}{p_0}\right)^{-\frac{1}{n_p}} = d_0 10^{\frac{p_{0,\text{dB}} - P_{\text{dB}}}{10 n_p}}. \tag{2.6}$$

The ML range estimator for the RSS problem is biased. This can be seen from the conditional moments of the log-normal P. Given $D = d$, it can be shown that the conditional moments for the RSS measurement are

$$\mathbb{E}[P^m | D = d] = p_0^m \left(\frac{d}{d_0}\right)^{-m n_p} \exp\left(\frac{1}{2}(n_p m b)^2\right), \tag{2.7}$$

where $b \equiv \frac{\sigma \ln 10}{10 n_p}$ as defined previously. From this it follows that

$$\mathbb{E}[\hat{D}_{\text{ML}}(P) | D = d] = \mathbb{E}\left[d_0 \left(\frac{P}{p_0}\right)^{-\frac{1}{n_p}} \middle| D = d\right] = d \exp\left(\frac{1}{2} b^2\right). \tag{2.8}$$

One can then form an unbiased estimator, by scaling the ML estimate. The resulting estimator is

$$\hat{D}_{\mathrm{UB}}(P) = d_0 \left(\frac{P}{p_0}\right)^{-\frac{1}{n_p}} \exp\left(-\frac{1}{2}b^2\right). \tag{2.9}$$

In conclusion, the ML range estimator and its unbiased (UB) estimator can be expressed as

$$\hat{d} = cd_0 \left(\frac{p}{p_0}\right)^{-\frac{1}{n_p}} \quad \text{where} \quad c = \begin{cases} 1 & \text{for ML} \\ \exp(-0.5b^2) & \text{for UB} \end{cases} \quad \text{and} \quad b \equiv \frac{\sigma \ln 10}{10 n_p}. \tag{2.10}$$

In other words,

$$\hat{d} = cd10^{-\frac{Z}{10n_p}}. \tag{2.11}$$

The mean square error (MSE) is a common statistics to measure the accuracy of estimator $\hat{\theta}$ for the fixed unknown parameter θ and it is defined by

$$\mathrm{MSE}(\hat{\theta}) \equiv \mathbb{E}[(\hat{\theta} - \theta)^2] = \mathrm{Var}(\hat{\theta}) + (\mathbb{E}[\hat{\theta}] - \theta)^2 \tag{2.12}$$

where the expectation is taken over all estimated parameters $\hat{\theta}$'s.

$$B(\hat{\theta}) \equiv \mathbb{E}[\hat{\theta}] - \theta \tag{2.13}$$

is called the bias and $\mathrm{MSE}(\hat{\theta}) = \mathrm{Var}(\hat{\theta}) + B^2(\hat{\theta})$. The root mean square error (RMSE) is the square root of the MSE, i.e., $\mathrm{RMSE}(\hat{\theta}) = \sqrt{\mathrm{MSE}(\hat{\theta})}$. Then the related first

Table 2.1: Statistics for Various Range Estimators

Method	Mean	Variance	MSE
ML	$0.5b^2d$	$\exp(b^2)\left[\exp(b^2)-1\right]d^2$	$\left[\exp(2b^2)-2\exp(0.5b^2)+1\right]d^2$
UB	d	$\left[\exp(b^2)-1\right]d^2$	$\left[\exp(b^2)-1\right]d^2$
CRB	d	b^2d^2	b^2d^2

Note: $b \equiv \sigma \ln 10/10/n_p$

Table 2.2: Illustrations of b and UB's c for Various n_p and Fixed $\sigma = 5$

	$n_p = 1.5$	$n_p = 2$	$n_p = 3$	$n_p = 5$
SNR=n_p/σ	0.3	0.4	0.6	1.0
$b \equiv \frac{\sigma \ln 10}{10n_p}$	0.7675	0.5756	0.3838	0.2303
$c \equiv \exp(-0.5b^2)$ for UB	0.7449	0.8473	0.9290	0.9738

and second order statistics for the ML-based estimator are as follows:

$$\mathbb{E}[\hat{d}|D=d] = cd\exp(0.5b^2), \tag{2.14}$$

$$\mathbb{E}[\hat{d}^2|D=d] = c^2d^2\exp(2b^2), \tag{2.15}$$

$$\mathrm{Var}(\hat{d}|D=d) = c^2d^2\exp(b^2)\left[\exp(b^2)-1\right], \tag{2.16}$$

$$\mathrm{MSE}(\hat{d}|D=d) = d^2\left[c^2\exp(2b^2)-2c\exp(0.5b^2)+1\right], \tag{2.17}$$

where $c = \exp(-1.5b^2)$ results in the lowest MSE for the ML-based range estimators. Finally, Table 2.1 summarizes the mean, variance, and MSE for ML range estimator, UB range estimator, and CRB.

Figure 2.1 shows how the RMSE of the various estimators depends on the actual

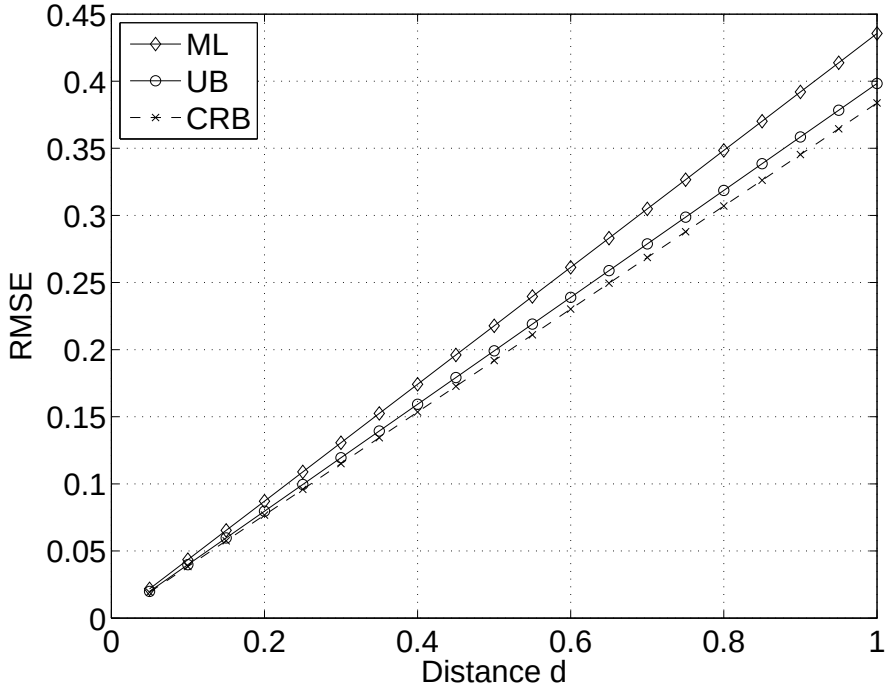

Figure 2.1: Range dependence of RMSE of various estimators with $n_p = 3$ and $\sigma = 5$.

range where $n_p = 3$ and $\sigma = 5$. For the ML-based estimators, the RMSEs increase linearly with the range as does the square root of Cramér-Rao bound. Moreover, the UB estimator has lower MSE than the ML one and the Cramér-Rao bound is unachievable by the UB estimator. As discussed before, the CRB and MSE depends on $b \equiv \frac{\sigma \ln 10}{10 n_p}$. Smaller values of n_p result in larger b which eventually increase error. Therefore, the quantity n_p/σ can be viewed as a SNR (signal to noise ratio) in the RSS localization problem. Figure 2.2 presents ML and UB range estimates with $\sigma = 5$ and various n_p. It demonstrates that the smaller SNR results in higher ranging error. Another observation shows that the gap between the ML and UB range estimates increases when the ratio n_p/σ decreases. Table 2.2 illustrates several b and scaling factor c for various n_p and fixed $\sigma = 5$. It can be found that c is closer

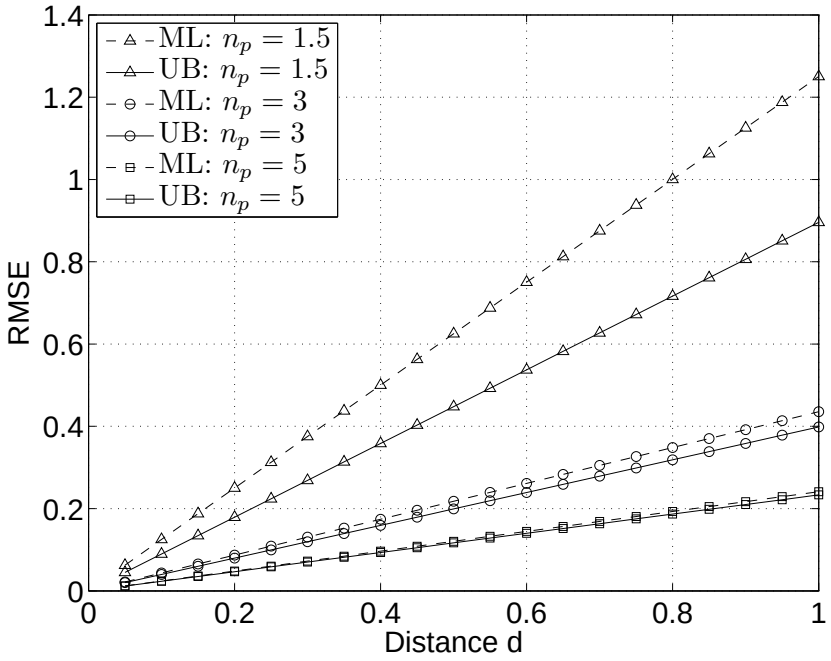

Figure 2.2: Range dependence of RMSE of UB and ML estimators with $\sigma = 5$ different n_p.

to 1 when the SNR increases and hence the larger gap for low SNR.

In conclusion, the accuracy of RSS ranging degrades when the actual range increases, which is different from TOA ranging. This characteristic inspires the proposed localization algorithm, which combines the estimates according to their distances. Furthermore, the lower n_p/σ decreases the ranging accuracy and thus increases the error of localization and limits the performance of many algorithms.

2.2 System Model and Maximum Likelihood Estimator

Because blindfolded nodes do not communicate with one another, non-cooperative localization can be viewed as a single blindfolded node that estimates its location by using only the received signals and locations of anchors. The locations of anchors are perfectly known and obviously need no further estimations. In addition, anchors

can be either fixed or mobile nodes. Besides anchors' locations, other parameters are also known. Thus the only estimation is the location of the blindfolded node.

The following discussion first applies to all dimensions and then emphasizes two dimensions for the purpose of illustration. The framework of two dimensions can be easily extended to other dimensions. Let $\Theta = [X, Y]$ be a random vector of the blindfolded node's unknown location and $\theta = [x, y]$ be one of its realization. The single unknown location can also be denoted by $\theta_0 \equiv \theta$, e.g., $[x_0, y_0] \equiv [x, y]$ in 2D to avoid confusion when discussing the proposed algorithm. Moreover, this is consistent to with notation that will be used in the context of cooperative localization. Then suppose that the blindfolded node can hear n anchors whose locations are at $\theta_1 = (x_1, y_1), \ldots, \theta_n = (x_n, y_n)$, respectively. Moreover, the distance between the blindfolded nodes and ith anchor is denoted by d_i and $d_i = \|\theta - \theta_i\|$ where $\|\cdot\|$ represents a norm. In this work, the norm used is the most common Euclidean norm (l_2-norm) and thus $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ in the 2-D case.

This dissertation mainly focuses on the received signal strength (RSS) model. The random powers are then governed by a log-normal distribution as in the range estimation and can be expressed as

$$P_i = p_0 \left(\frac{d_i}{d_0}\right)^{-n_p} 10^{\frac{Z_i}{10}} \tag{2.18}$$

for i from 1 to n. All above parameters are defined as before and the Z_i are independent and identically distributed (i.i.d.) zero mean Gaussian random variables with variance σ^2. Let $\rho_i = p_{0,\mathrm{dB}} - 10n_p \log_{10}(\frac{d_i}{d_0})$, and then the joint conditional PDF can be expressed as

$$f_{\mathbf{P}_{\mathrm{dB}}}(\mathbf{p}_{\mathrm{dB}}|\Theta = \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(p_{i,\mathrm{dB}} - \rho_i)^2}{2\sigma^2}\right) \tag{2.19}$$

The maximum likelihood estimator (MLE) in the RSS non-cooperative localization can be written as

$$\begin{aligned}\hat{\theta}_{\mathrm{ML}}(\mathbf{p}) &= \underset{\vartheta\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\log\left(\frac{p_i}{p_0}\left(\frac{\|\vartheta-\theta_i\|}{d_0}\right)^{n_p}\right)\right]^2 \\ &= \underset{\vartheta\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\log\left(\frac{d(\vartheta,\theta_i)}{\hat{d}_{i,\mathrm{ML}}}\right)\right]^2 \end{aligned} \tag{2.20}$$

where $\mathbf{p} = [p_1, \ldots, p_n]$ is a vector containing the received power measurements from the anchors and $\mathcal{S}$ represents a search space. Moreover, $\hat{d}_{i,\mathrm{ML}} = d_0(p_i/p_0)^{-1/n_p}$ is the ML range estimate and $d(\vartheta, \theta_i) = \|\vartheta - \theta_i\|$ represents the distance between a candidate ϑ and the i-th anchor θ_i. In two dimensions, the MLE can be written as:

$$\begin{aligned}\hat{\theta}_{\mathrm{ML}}(\mathbf{p}) &= \underset{(\xi,\psi)\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\log\left(\frac{p_i}{p_0}\left(\frac{\sqrt{(\xi-x_i)^2+(\psi-y_i)^2}}{d_0}\right)^{n_p}\right)\right]^2 \\ &= \underset{(\xi,\psi)\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=1}^{n} \left[\log\left(\frac{d([\xi,\psi],[x_i,y_i])}{\hat{d}_{i,\mathrm{ML}}}\right)\right]^2 \end{aligned} \tag{2.21}$$

where $d([\xi,\psi],[x_i,y_i]) = \sqrt{(\xi-x_i)^2+(\psi-y_i)^2}$. Besides the received powers and anchors' locations, the MLE requires the channel parameters n_p and $p_0(d_0)$. Since the above cost function is non-convex, the estimate by gradient methods may stay in a local extreme point. Moreover, the parameters and constraints of gradient method should carefully choose. For example, the gradient descent with Armijo rule [43] can have reasonable result by setting a range of acceptable estimates and parameters. To avoid local minima and adjustment of parameters for different wireless environments, an exhaustive search (brute-force search) in a discrete search space is applied to MLE.

One may consider the case where the anchors' locations and the channel parameters are not perfectly known. The cost function in (2.20) is still identical, but

the search space is much larger and the exhaustive search becomes infeasible. Section 2.8 will discuss how to estimate the channel parameters by utilizing the signals and locations from anchors.

2.3 Linear Combination Estimator

This section introduces a new location estimator, which linearly combines the multiple ML-based location estimates determined by the signals and locations of different anchors. The new algorithm mainly contributes to assign combining weights according to relative distances based on the concept of the best linear unbiased estimator (BLUE) [41]. Because of the simple ML-based range estimators in (2.10), this linear combination (LC) estimator should decrease the computation time and/or memory consumption in the MLE, especially when the number of measurements increases. Chan at el. proposed another BLUE type algorithm used in the TOA localization [44]. Besides the different measurement model, the main difference is that their method solves the least-squares equations and requires no iterations.

A two-dimensional problem is considered first and other dimensional cases can be extended using the general expression. The unknown location of a single blindfolded node $\theta_0 \equiv \theta$ as mentioned previously, e.g., $[x_0, y_0] \equiv [x, y]$ in 2D. The location estimate, derived from the anchor i, is denoted by $\hat{\theta}_{i0}$ by utilizing the anchor's location $\theta_i = [x_i, y_i]$, the range estimate $\hat{d}_i$ between the anchor and the blindfolded node, and the corresponding direction $\mathbf{v}_i$ as illustrated in Figure 2.3. More explicitly,

$$\hat{\theta}_{i0} = \theta_i + \hat{d}_i \mathbf{v}_i = \theta_i + \hat{d}_i \frac{\theta - \theta_i}{\|\theta - \theta_i\|}, \tag{2.22}$$

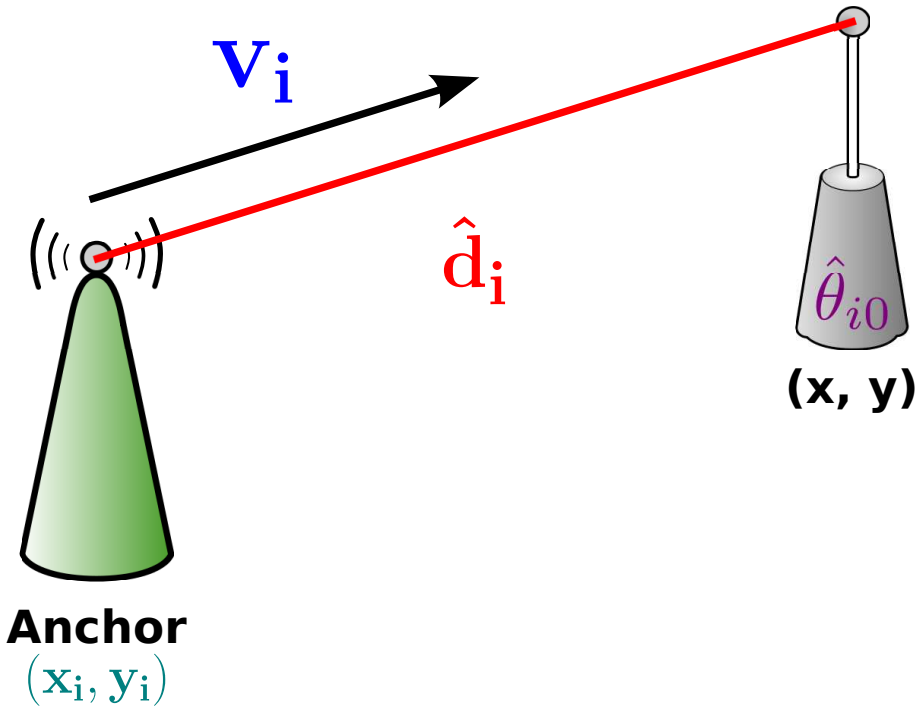

Figure 2.3: Illustration of a location estimate derived from anchor i.

and it can be expressed in two dimensions as

$$\hat{\theta}_{i0} = [x_i, y_i] + \hat{d}_i \mathbf{v}_i = [x_i, y_i] + \hat{d}_i \frac{[x - x_i, y - y_i]}{\sqrt{(x - x_i)^2 + (y - y_i)^2}}. \tag{2.23}$$

If an unbiased range estimator is applied, i.e., $\mathbb{E}[\hat{d}_i] = d_i$,

$$\mathbb{E}[\hat{\theta}_{i0}] = \theta_i + \mathbb{E}[\hat{d}_i] \frac{\theta - \theta_i}{d_i} = \theta. \tag{2.24}$$

In other words, $\hat{\theta}_{i0}$ is also unbiased given an unbiased range estimate and the correct direction.

Applying the Gauss-Markov theorem, these n unbiased location estimates can form the best linear unbiased estimator (BLUE) [41] for $\hat{\theta}_{\text{opt}} = \sum_{i=1}^{n} a_{i,\text{opt}} \hat{\theta}_{i0}$. The

optimal weight vector is

$$\mathbf{a}_{\text{opt}} = [a_{1,\text{opt}} \cdots a_{n,\text{opt}}]^T = \frac{\mathbf{C}^{-1}\mathbf{s}}{\mathbf{s}^T\mathbf{C}^{-1}\mathbf{s}} \tag{2.25}$$

where $\mathbf{s} = [1 \cdots 1]^T$ is the scaled mean vector and $\mathbf{C}$ is the covariance matrix of the location estimates where $\mathbf{C}_{ij} = \mathbb{E}[(\hat{\theta}_{i0} - \theta)(\hat{\theta}_{j0} - \theta)]$. It is noticed that $0 \leq a_{i,\text{opt}} \leq 1$ and $\sum a_{i,\text{opt}} = 1$.

To make the variance of a vector to be a scalar, define

$$\text{Var}(\hat{\theta}_{i0}) \equiv \mathbb{E}\left[\left\|\hat{\theta}_{i0} - \mathbb{E}[\hat{\theta}_{i0}]\right\|^2\right] = \mathbb{E}\left[\left\|\hat{\theta}_{i0} - \theta\right\|^2\right], \tag{2.26}$$

and for the 2-D case, the variance becomes

$$\text{Var}(\hat{\theta}_{i0}) = \mathbb{E}\left[(\hat{x}_{i0} - x)^2 + (\hat{y}_{i0} - y)^2\right] = \mathbb{E}[\hat{x}_{i0}^2] - x^2 + \mathbb{E}[\hat{y}_{i0}^2] - y^2 \tag{2.27}$$

where

$$\begin{aligned}
\mathbb{E}[\hat{x}_{i0}^2] &= \mathbb{E}\left[\left(x_i + \hat{d}_i\frac{x - x_i}{d_i}\right)^2\right] \\
&= x_i^2 + \mathbb{E}[\hat{d}_i^2]\frac{(x - x_i)^2}{d_i^2} + 2x_i\mathbb{E}[\hat{d}_i]\frac{x - x_i}{d_i} \\
&= x_i^2 + \mathbb{E}[\hat{d}_i^2]\frac{(x - x_i)^2}{d_i^2} + 2x_i x - 2x_i^2, \qquad (2.28) \\
\mathbb{E}[\hat{y}_{i0}^2] &= y_i^2 + \mathbb{E}[\hat{d}_i^2]\frac{(y - y_i)^2}{d_i^2} + 2y_i y - 2y_i^2. \qquad (2.29)
\end{aligned}$$

Thus the variance of estimate utilizing anchor i in 2D is

$$\begin{aligned}\mathrm{Var}(\hat{\theta}_{i0}) &= \mathbb{E}[\hat{d}_i^2] + 2x_i x + 2y_i y - x_i^2 - y_i^2 - x^2 - y^2 \\ &= \mathbb{E}[\hat{d}_i^2] - (x - x_i)^2 - (y - y_i)^2 \\ &= \mathbb{E}[\hat{d}_i^2] - d_i^2 = \mathrm{Var}(\hat{d}_i) = [\exp(b^2) - 1]d_i^2. \end{aligned} \tag{2.30}$$

$\mathrm{Var}(\hat{\theta}_{i0}) = \mathrm{Var}(\hat{d}_i)$ is still true for other dimensions with unbiased range estimates and known directions. Since $\mathrm{Var}(\hat{\theta}_{i0}) = \mathrm{Var}(\hat{d}_i)$ and the power measurements, p_i, are independent, the covariance matrix becomes $\mathbf{C} = \mathrm{diag}(d_1^2[\exp(b^2)-1], \ldots, d_n^2[\exp(b^2)-1])$. Finally,

$$a_{i,\mathrm{opt}} = \frac{1/d_i^2}{\sum_{j=1}^{n} 1/d_j^2} \quad \text{for} \quad i = 1, \ldots, n, \tag{2.31}$$

$$\mathrm{MSE}(\hat{\theta}_{\mathrm{opt}}) = \mathrm{Var}(\hat{\theta}_{\mathrm{opt}}) = \frac{1}{\mathbf{s}^T \mathbf{C}^{-1} \mathbf{s}} = \frac{\exp(b^2) - 1}{\sum_{j=1}^{n} 1/d_j^2}. \tag{2.32}$$

It is found that the weights are higher for the anchors closer to the blindfolded node, because the error of the range estimate increases with the range in the RSS model as shown in Table 2.1 and Figure 2.1. This property in weights disappears in the TOA model because the range error is independent of the distance. The weights are all equal even the anchors have different distances to the blindfolded node. Moreover, the MLE is the same as the LC estimator in TOA and there is no advantage for the LC estimator. In addition, the variances of both the unbiased range estimator and the LC estimator with true directions have the same scaling factor $\exp(b^2) - 1$.

The correct distances are unavailable to optimal weights $\mathbf{a}_{\mathrm{opt}}$'s. Thus the estimated ranges $\hat{d}_i$'s are used to compute the combining weights a_i's. Moreover, the true directions require the correct location of the blindfolded node. Estimated directions therefore are obtained by iteratively refining the location estimate of the blindfold

node. In summary, the procedure of the LC estimator for a single unknown location is as follows:

1. Calculate the UB range estimates $\hat{d}_i$ in (2.10)

$$\hat{d}_i = \exp(-0.5b^2) d_0 \left(\frac{p_i}{p_0}\right)^{-\frac{1}{n_p}}, \tag{2.33}$$

and the combining weights

$$a_i = \frac{1/\hat{d}_i^2}{\sum_{j=1}^{n} 1/\hat{d}_j^2} \quad \text{for} \quad i = 1, \ldots, n. \tag{2.34}$$

2. Assign an arbitrary location as the initial location $[\hat{x}^{(0)}, \hat{y}^{(0)}]$ and let $k = 0$.

3. Compute the normalized directions (sine and cosine) from the anchors to the node:

$$\mathbf{v}_i^{(k)} = \frac{\hat{\theta}^{(k)} - \theta_i}{\|\hat{\theta}^{(k)} - \theta_i\|} = \frac{[\hat{x}^{(k)} - x_i, \hat{y}^{(k)} - y_i]}{\|[\hat{x}^{(k)} - x_i, \hat{y}^{(k)} - y_i]\|} \quad \text{for} \quad i = 1, \ldots, n \tag{2.35}$$

if $\|\hat{\theta}^{(k)} - \theta_i\| = \left\|[\hat{x}^{(k)} - x_i, \hat{y}^{(k)} - y_i]\right\| \neq \mathbf{0}$; $\mathbf{v}_i^{(k)} = \mathbf{0}$, otherwise (it makes 0/0 as 0). $\|\cdot\|$ represents the Euclidean norm (l_2-norm).

4. The new location estimate in the $(k+1)$-th stage is

$$\hat{\theta}^{(k+1)} = \left[\hat{x}^{(k+1)}, \hat{y}^{(k+1)}\right] = \sum_{i=1}^{n} a_i \left(\theta_i + \hat{d}_i \mathbf{v}_i^{(k)}\right) = \sum_{i=1}^{n} a_i \left([x_i, y_i] + \hat{d}_i \mathbf{v}_i^{(k)}\right). \tag{2.36}$$

5. Stop the procedure if

$$\frac{\left\|\hat{\theta}^{(k+1)} - \hat{\theta}^{(k)}\right\|^2}{\left\|\hat{\theta}^{(k)}\right\|^2} = \frac{\left(\hat{x}^{(k+1)} - \hat{x}^{(k)}\right)^2 + \left(\hat{y}^{(k+1)} - \hat{y}^{(k)}\right)^2}{\left(\hat{x}^{(k)}\right)^2 + \left(\hat{y}^{(k)}\right)^2} < \epsilon \tag{2.37}$$

or

$$k > \text{maximum number of iterations.} \tag{2.38}$$

Otherwise, let $k = k + 1$ and then repeat step 3) $-$ 5).

2.4 Analysis of Estimation Error

The lower bound on the estimation error for non-cooperative LC was derived while developing the algorithm with true directions as shown in (2.32). This section investigates an upper bound and an approximation by using randomly guessed directions in two dimensions. The approximation is computed by knowing the optimal weights and is actually an upper bound for the LC estimator with optimal weights. It is called an approximation mainly because the LC algorithm in Section 2.3 and the LC with optimal weights have very similar MSE. The upper bound and the approximation will be compared with the lower bound and the actual LC performance throuhg simulations in the next subsection. A system designer can then deploy a certain number of anchors with designed locations or location distribution according to two bounds and an approximation to achieve the desired accuracy.

Let $\theta = [x, y]$ be the true location of the blindfold node. Again, let $\theta \equiv \theta_0 = [x_0, y_0]$ to avoid ambiguity. The estimated direction between anchor i and the blindfolded node is denoted by $[\cos(\hat{\phi}_i), \sin(\hat{\phi}_i)]$. Then, the location estimate from anchor i is given by

$$\hat{\theta}_{i0} = [\hat{x}_{i0}, \hat{y}_{i0}] = [x_i, y_i] + \hat{d}_i[\cos(\hat{\phi}_i), \sin(\hat{\phi}_i)]. \tag{2.39}$$

The true location θ can be viewed from anchor i as

$$\theta = [x, y] = [x_i, y_i] + d_i[\cos(\phi_i), \sin(\phi_i)] \tag{2.40}$$

by replacing the estimated range and angle with the true ones.

The MSE of a linear combination estimator with $\sum a_i = 1$ can be written as

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}_{\hat{\theta}}[\|\hat{\theta} - \theta\|^2] = \mathbb{E}\left[\left\|\sum_{i=1}^{n} a_i\hat{\theta}_{i0} - \theta\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^{n} a_i(\hat{\theta}_{i0} - \theta)\right\|^2\right]. \tag{2.41}$$

In the 2-D case, the MSE becomes

$$\begin{aligned}
\mathrm{MSE}(\hat{\theta}) &= \mathbb{E}\left[\left\|\sum_{i=1}^{n} a_i[\hat{x}_{i0} - x, \hat{y}_{i0} - y]\right\|^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} a_i(\hat{x}_{i0} - x)\right)^2 + \left(\sum_{i=1}^{n} a_i(\hat{y}_{i0} - y)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} a_i[(\hat{x}_{i0} - x)(\hat{x}_{j0} - x) + (\hat{y}_{i0} - y)(\hat{y}_{j0} - y)]a_j\right] \\
&\equiv \mathbb{E}[\mathbf{a}^{\mathrm{T}}\mathbf{M}\mathbf{a}]
\end{aligned} \tag{2.42}$$

where $\mathbf{a} = [a_1 \cdots a_n]^{\mathrm{T}}$ and $\mathbf{M}$ is an n by n matrix. If the weights are independent of location estimates, they are no longer random variables and

$$\mathrm{MSE}(\hat{\theta}) = \mathbf{a}^{\mathrm{T}}\mathbb{E}[\mathbf{M}]\mathbf{a} = \sum_{i=1}^{n} a_i^2\mathbb{E}[\mathbf{M}_{ii}] + \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} a_i a_j \mathbb{E}[\mathbf{M}_{ij}]. \tag{2.43}$$

In addition, the elements of matrix $\mathbf{M}_{ij}$ $(i \neq j)$ and $\mathbf{M}_{ii}$ can be simplified to

$$\begin{aligned}
\mathbf{M}_{ij} &= (\hat{d}_i \cos\hat{\phi}_i - d_i \cos\phi_i)(\hat{d}_j \cos\hat{\phi}_j - d_j \cos\phi_j) \\
&\quad + (\hat{d}_i \sin\hat{\phi}_i - d_i \sin\phi_i)(\hat{d}_j \sin\hat{\phi}_j - d_j \sin\phi_j) \quad \text{for} \quad i \neq j, \qquad (2.44) \\
\mathbf{M}_{ii} &= \hat{d}_i^2 \cos^2\hat{\phi}_i + d_i^2 \cos^2\phi_i - 2\hat{d}_i d_i \cos\hat{\phi}_i \cos\phi_i \\
&\quad + \hat{d}_i^2 \sin^2\hat{\phi}_i + d_i^2 \sin^2\phi_i - 2\hat{d}_i d_i \sin\hat{\phi}_i \sin\phi_i \\
&= \hat{d}_i^2 + d_i^2 - 2\hat{d}_i d_i(\cos\hat{\phi}_i \cos\phi_i + \sin\hat{\phi}_i \sin\phi_i) \qquad (2.45)
\end{aligned}$$

The proposed LC algorithm uses the previous estimated location to estimate angle $\hat{\phi}_i$ as in (2.35). The plain case of angle estimation is a uniform guess over $[0, 2\pi)$ which means that the estimator has no other information. If any method of angle estimation has a larger error than the random guess under certain environment, it should be replaced by the random guess to ensure the worse case valid. This requires simulations or field test before executing the LC estimator and might not be available. To derivate upper bounds, the cases of angel error worse than the error of random guess are assumed to be known in advance and the angle estimations can be replaced by the random guess angle. The random guess makes $\hat{\phi}_i$ independent of $\hat{d}_i$, $\hat{\phi}_j$, and $\hat{d}_j$. Then

$$\begin{aligned}
\mathbb{E}[\mathbf{M}_{ij}] &= \left(\mathbb{E}[\hat{d}_i]\mathbb{E}[\cos\hat{\phi}_i] - d_i \cos\phi_i\right)\left(\mathbb{E}[\hat{d}_j]\mathbb{E}[\cos\hat{\phi}_j] - d_j \cos\phi_j\right) \\
&\quad + \left(\mathbb{E}[\hat{d}_i]\mathbb{E}[\sin\hat{\phi}_i] - d_i \sin\phi_i\right)\left(\mathbb{E}[\hat{d}_j]\mathbb{E}[\sin\hat{\phi}_j] - d_j \sin\phi_j\right), \qquad (2.46) \\
\mathbb{E}[\mathbf{M}_{ii}] &= \mathbb{E}[\hat{d}_i^2] + d_i^2 - 2\mathbb{E}[\hat{d}_i] d_i(\mathbb{E}[\cos\hat{\phi}_i]\cos\phi_i + \mathbb{E}[\sin\hat{\phi}_i]\sin\phi_i) \qquad (2.47)
\end{aligned}$$

Moreover, $\mathbb{E}[\cos\hat{\phi}_i] = 0$ and $\mathbb{E}[\sin\hat{\phi}_i] = 0$ when $\hat{\phi}_i$ is uniformly distributed in an interval $[0, 2\pi)$. Applying the statistics of unbiased range estimates: $\mathbb{E}[\hat{d}_i] = d_i$ and

$\mathbb{E}[\hat{d}_i^2] = d_i^2 \exp(b^2)$, the expectation of the matrix $\mathbf{M}$ becomes

$$\mathbb{E}[\mathbf{M}_{ij}] = d_i d_j (\cos(\phi_i)\cos(\phi_j) + \sin(\phi_i)\sin(\phi_j)) \quad \text{for} \quad i \neq j, \tag{2.48}$$

$$\mathbb{E}[\mathbf{M}_{ii}] = d_i^2 \left[\exp(b^2) + 1\right]. \tag{2.49}$$

Therefore, the MSE of LC estimator with optimal weights $a_{i,\text{opt}} = \frac{1/d_i^2}{\sum_{k=1}^n 1/d_k^2}$ that are not random variables can written as

$$\begin{aligned}
\text{MSE}(\hat{\theta}) &= \sum_{i=1}^n a_{i,\text{opt}}^2 \mathbb{E}[\mathbf{M}_{ii}] + \sum_{i=1}^n \sum_{j=1, j\neq i}^n a_{i,\text{opt}} a_{j,\text{opt}} \mathbb{E}[\mathbf{M}_{ij}] \qquad (2.50)\\
&\leq \sum_{i=1}^n \frac{\left(\frac{1}{d_i^2}\right)^2 d_i^2 \left[\exp(b^2)+1\right]}{(\sum_{k=1}^n 1/d_k^2)^2} + \sum_{i=1}^n \sum_{j=1, j\neq i}^n \frac{\frac{1}{d_i^2 d_j^2} d_i d_j \cos(\phi_i - \phi_j)}{(\sum_{k=1}^n 1/d_k^2)^2} \\
&= \frac{[\exp(b^2)+1]\sum_{i=1}^n \frac{1}{d_i^2}}{(\sum_{k=1}^n 1/d_k^2)^2} + \frac{\sum_{i=1}^n \sum_{j=1, j\neq i}^n \frac{\cos(\phi_i - \phi_j)}{d_i d_j}}{(\sum_{k=1}^n 1/d_k^2)^2} \\
&= \frac{\exp(b^2)+1}{\sum_{k=1}^n 1/d_k^2} + \frac{\sum_{i=1}^n \sum_{j=1, j\neq i}^n \frac{\cos(\phi_i - \phi_j)}{d_i d_j}}{(\sum_{k=1}^n 1/d_k^2)^2}. \qquad (2.51)
\end{aligned}$$

which provides the upper bound on the non-cooperative linear combination estimator with optimal weights. In the next section, the simulations will reveal this LC with optimal weights and its upper bound will be closer as the SNR is lower. Moreover, the simulations illustrate that the MSE of the LC estimator in reality, whose weights are $a_i = \frac{1/\hat{d}_i^2}{\sum_{k=1}^n 1/\hat{d}_k^2}$, is closer to the MSE of the LC with optimal weights. Neither methods outperform the other in every situation (LC: $\mathbf{a}_{\text{opt}}$ ensures the lowest MSE when true directions are provided) and the random guess could be better than the proposed iterative angle estimation in low SNR regime. Thus the above result in (2.51) is called the "approximation" of LC estimators although its MSE is higher than the MSE of LC estimators in most cases. Compared with the lower bound in

(2.32):

$$\mathrm{MSE}(\hat{\theta}) \geq \frac{\exp(b^2) - 1}{\sum_{k=1}^{n} 1/d_k^2}, \tag{2.52}$$

the approximation has a larger scaling factor and additional terms from non-diagonal exponents.

Instead of the optimal weight, the equal weight ($a_i = 1/n$, not random) combines with the random angle guess to provide the upper bound as following:

$$\mathrm{MSE}(\hat{\theta}) \leq \sum_{i=1}^{n} \frac{d_i^2 \left[\exp(b^2) + 1\right]}{n^2} + \sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \frac{d_i d_j \cos(\phi_i - \phi_j)}{n^2} \tag{2.53}$$

As mentioned before, this result is valid for fixed combing weights. The LC estimator in reality, whose weights are $a_i = \frac{1/\hat{d}_i^2}{\sum_{k=1}^{n} 1/\hat{d}_k^2}$, are also random variables. However, this upper bound is loose and the RMSE of LC estimator is much lower according to all simulations in the next section. Therefore, (2.53) is named an upper bound of estimation error on the non-cooperative LC.

To derive the variance under perfect weights and random angle estimations, the expectation of the estimate with fixed weights should be computed first. That is,

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}\left[\sum_{i=1}^{n} a_i \hat{\theta}_{i0}\right] = \sum_{i=1}^{n} a_i \mathbb{E}[\hat{\theta}_{i0}] = \sum_{i=1}^{n} a_i \theta_i \tag{2.54}$$

which depends on anchors locations and weight. It can be simplified as $\sum_{i=1}^{n} \theta_i/n$ for equal-weight cases. The variance of a linear combination estimator in the above situation can be written as

$$\mathrm{Var}(\hat{\theta}) = \mathbb{E}_{\hat{\theta}}[\|\hat{\theta} - \mathbb{E}[\hat{\theta}]\|^2] = \mathbb{E}\left[\left\|\sum_{i=1}^{n} a_i(\hat{\theta}_{i0} - \theta_i)\right\|^2\right]. \tag{2.55}$$

The variance for two dimensions becomes

$$\begin{aligned}
\mathrm{Var}(\hat{\theta}) &= \mathbb{E}\left[\left\|\sum_{i=1}^{n} a_i[\hat{x}_{i0}-x_i, \hat{y}_{i0}-y_i]\right\|^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{n} a_i(\hat{x}_{i0}-x_i)\right)^2 + \left(\sum_{i=1}^{n} a_i(\hat{y}_{i0}-y_i)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} a_i[(\hat{x}_{i0}-x)(\hat{x}_{j0}-x)+(\hat{y}_{i0}-y)(\hat{y}_{j0}-y)]a_j\right] \\
&\equiv \mathbb{E}[\mathbf{a}^{\mathrm{T}}\mathbf{V}\mathbf{a}] = \mathbf{a}^{\mathrm{T}}\mathbb{E}[\mathbf{V}]\mathbf{a}. \qquad (2.56)
\end{aligned}$$

where $\mathbf{a} = [a_1 \cdots a_n]^{\mathrm{T}}$ and $\mathbf{V}$ is an n by n matrix. $\mathbf{V}_{ij}$ and $\mathbf{V}_{ii}$ can be expressed as

$$\begin{aligned}
\mathbf{V}_{ij} &= \hat{d}_i\hat{d}_j\cos\hat{\phi}_i\cos\hat{\phi}_j + \hat{d}_i\hat{d}_j\sin\hat{\phi}_i\sin\hat{\phi}_j, \qquad (2.57)\\
\mathbf{V}_{ii} &= \hat{d}_i^2\cos^2\hat{\phi}_i + \hat{d}_i^2\sin^2\hat{\phi}_i = \hat{d}_i^2. \qquad (2.58)
\end{aligned}$$

Again, applying $\mathbb{E}[\cos\hat{\phi}_i] = 0$, $\mathbb{E}[\sin\hat{\phi}_i] = 0$, and $\mathbb{E}[\hat{d}_i^2] = d_i^2\exp(b^2)$ for randomly guessed angles, the expected values of elements in the matrix are bounded as

$$\begin{aligned}
\mathbb{E}[\mathbf{V}_{ij}] &= 0, \qquad (2.59)\\
\mathbb{E}[\mathbf{V}_{ii}] &= d_i^2\exp(b^2). \qquad (2.60)
\end{aligned}$$

and $\mathbb{E}[\mathbf{V}]$ thus is diagonal. Finally, an approximation of variance can be derived by assigning the optimal weight:

$$\begin{aligned}
\mathrm{Var}(\hat{\theta}) &\le \sum_{i=1}^{n} a_{i,\mathrm{opt}}^2\mathbb{E}[\mathbf{V}_{ii}] = \sum_{i=1}^{n}\frac{\left(\frac{1}{d_i^2}\right)^2 d_i^2\exp(b^2)}{(\sum_{k=1}^{n}1/d_k^2)^2} \\
&= \frac{\exp(b^2)\sum_{i=1}^{n}\frac{1}{d_i^2}}{(\sum_{k=1}^{n}1/d_k^2)^2} = \frac{\exp(b^2)}{\sum_{k=1}^{n}1/d_k^2} \qquad (2.61)
\end{aligned}$$

On the other hand, the variance for equal combining weights ($a_i = 1/n$) and random angle estimations is

$$\mathrm{Var}(\hat{\theta}) \leq \sum_{i=1}^{n} a_i^2 \mathbb{E}[\mathbf{V}_{ii}] = \sum_{i=1}^{n} \frac{d_i^2 \exp(b^2)}{n^2} \tag{2.62}$$

which provides the upper bound on the variance of the non-cooperative linear combination estimator. The lower bound of the variance is the same as the lower bound of the MSE because it is unbiased, i.e., $\mathrm{Var}(\hat{\theta}_{\mathrm{opt}}) = [\exp(b^2) - 1]/(\sum_{k=1}^{n} 1/d_k^2)$.

If anchors' locations are also random, the distance and direction become random variables and can be denoted by D_i and Φ_i, respectively. The previous case with fixed anchors' locations can be viewed as conditioning on $D_i = d_i$ and $\Phi_i = \phi_i$. Then the MSE needs to be averaged over $\hat{\theta}$, $D \equiv [D_1, \ldots, D_n]$, and $\Phi \equiv [\Phi_1, \ldots, \Phi_n]$, i.e.,

$$\mathrm{MSE}(\hat{\theta}) = \mathbb{E}_{\hat{\theta} D \Phi}[\|\hat{\theta} - \theta\|^2]. \tag{2.63}$$

The upper bound using equal weight and random angles becomes

$$\mathrm{MSE}(\hat{\theta}) \leq \mathbb{E}_D \left[\sum_{k=1}^{n} \frac{D_k^2[\exp(b^2) + 1]}{n^2} \right] + \mathbb{E}_{D\Phi} \left[\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \frac{D_i D_j \cos(\Phi_i - \Phi_j)}{n^2} \right]. \tag{2.64}$$

If the directions from the anchors to the blindfolded node are uniformly distributed, e.g., anchors are randomly placed inside a circle, the non-diagonal term will be 0 because $\mathbb{E}_{\Phi_i \Phi_j}[\cos(\Phi_i - \Phi_j)] = 0$. Then the upper bound becomes

$$\mathrm{MSE}(\hat{\theta}) \leq \frac{\exp(b^2) + 1}{n^2} \sum_{k=1}^{n} \mathbb{E}_D \left[D_k^2 \right]. \tag{2.65}$$

Similarly, the approximation by knowing the optimal weights now is

$$\mathrm{MSE}(\hat{\theta}) \leq \mathbb{E}_D \left[\frac{\exp(b^2)+1}{\sum_{k=1}^{n} 1/D_k^2} \right] + \mathbb{E}_{D\Phi} \left[\frac{\sum_{i=1}^{n} \sum_{j=1, j\neq i}^{n} \frac{\cos(\Phi_i - \Phi_j)}{D_i D_j}}{(\sum_{k=1}^{n} 1/D_k^2)^2} \right]. \tag{2.66}$$

Again, $\mathbb{E}_{\Phi_i \Phi_j}[\cos(\Phi_i - \Phi_j)] = 0$ if the directions are uniformly distributed. Then the approximation that becomes

$$\mathrm{MSE}(\hat{\theta}) \leq \mathbb{E} \left[\frac{\exp(b^2)+1}{\sum_{j=1}^{n} 1/D_j^2} \right]. \tag{2.67}$$

The lower bound is given by

$$\mathrm{MSE}(\hat{\theta}) \geq \mathbb{E} \left[\frac{\exp(b^2)-1}{\sum_{j=1}^{n} 1/D_j^2} \right] \tag{2.68}$$

when knowing the directions. The lower bound and the approximation are only different in constants and knowing $\mathbb{E}\left[\frac{1}{\sum_{j=1}^{n} 1/D_j^2}\right]$ can quantify the LC accuracy. They are especially important when the topologies are random. There is therefore no need to run many trails for each topology to obtain MSE with respect to noisy power measurements.

2.5 Examples and Simulation Results

The MLE and variations of LC estimators including bounds in two 2-D topological examples and the related simulation results of accuracy are provided in this subsection. Another important performance measure – the number of iterations will be discussed in Section 2.6 because it determines computation cost. In the first example, a blindfolded node is inside a unit square and anchors are at the four corners. The second example considers a blindfolded node at the origin and numbers of anchors are randomly placed inside the unit circle with the centre at the origin.

2.5.1 Unit Square

In the first example, the observed true locations of the blindfolded node start from (0.05, 0.05), and the x or y coordinate is increased by 0.05 until (0.95, 0.95). Thus there are totally $19 \times 19 = 361$ observed locations. In the MLE, a step size is 0.005 which generates a discrete search space $\mathcal{S} = \{0, 0.005, \ldots, 0.995, 1\} \times \{0, 0.005, \ldots, 0.995, 1\}$ and it results in finding the minimum from 4×10^4 candidates. The RMSE difference between step size = 0.005 and infinitely small step size is at most 0.0035 ($\sqrt{2(0.005/2)^2}$) if the cost function is smooth near the estimated location. In the LC, the stopping ϵ is set to be 10^{-8}, and the permitted maximum number of iterations is 10^4. In addition to the conventional LC, the bounded LC (BLC) where the final estimate is restricted to the nearest point within the boundaries, in comparison with the constrained search space in the MLE. Unlike MLE, the LC uses unbiased range estimates, which require knowledge of the channel parameter σ. If σ is unavailable, the UB range estimates are replaced by the pure ML range estimates to provide the "LC: ML ranges" estimator. The bounded LC using ML ranges called "BLC: ML ranges" is the LC estimator to be fairly compared with the MLE, because both use pure ML ranges and are bounded by four sides of unit square. Thus 4 different LCs and the MLE are investigated.

The accuracy of non-cooperative localizations is measured by the following quantities. The MSE for a fixed observed location θ is defined as

$$\mathrm{MSE}(\hat{\theta}) \equiv \mathbb{E}[\|\hat{\theta} - \theta\|^2] \tag{2.69}$$

where the expectation is taken over all $\hat{\theta}$ and the RMSE is the square root of MSE.

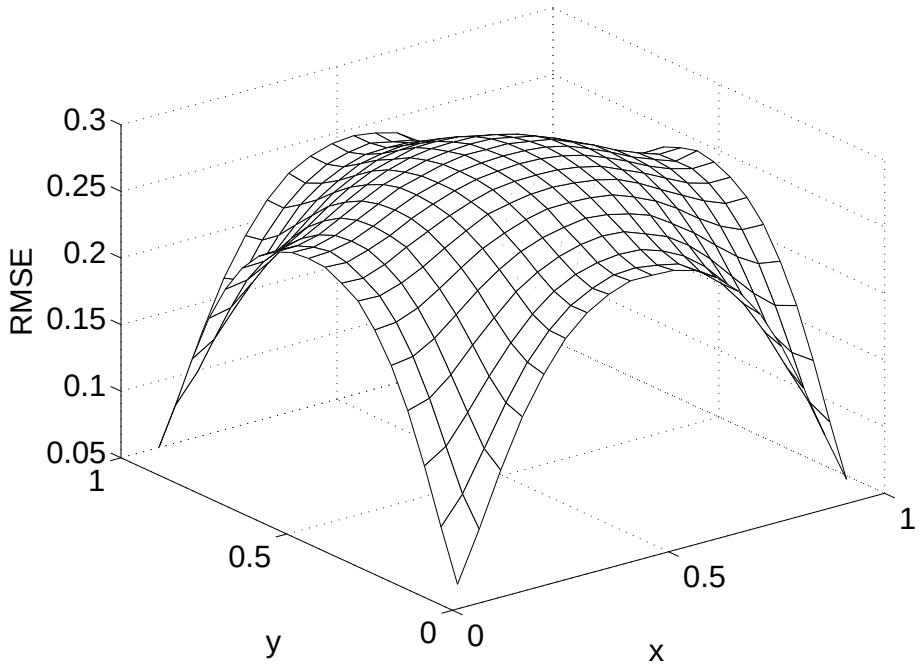

Figure 2.4: RMSE of the maximum likelihood estimator versus the coordinates of one blindfolded node given $n_p = 3$ and $\sigma = 5$.

In the simulation, sample RMSE is used and given by

$$\text{(Sample) RMSE}(\hat{\theta}) \equiv \sqrt{\frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} \|\hat{\theta}_i - \theta\|^2} \tag{2.70}$$

where $\hat{\theta}_i$ is the estimation result of the i-th trial and N_{sim} is the total number of trials. The word "sample" will be omitted for simplicity. The bias is as defined in (2.13) but it could be a multiple-component vector. Thus a measure "bias length" is given as

$$\text{bias length} \equiv \|B(\hat{\theta})\| = \|\mathbb{E}[\hat{\theta}] - \theta\| \tag{2.71}$$

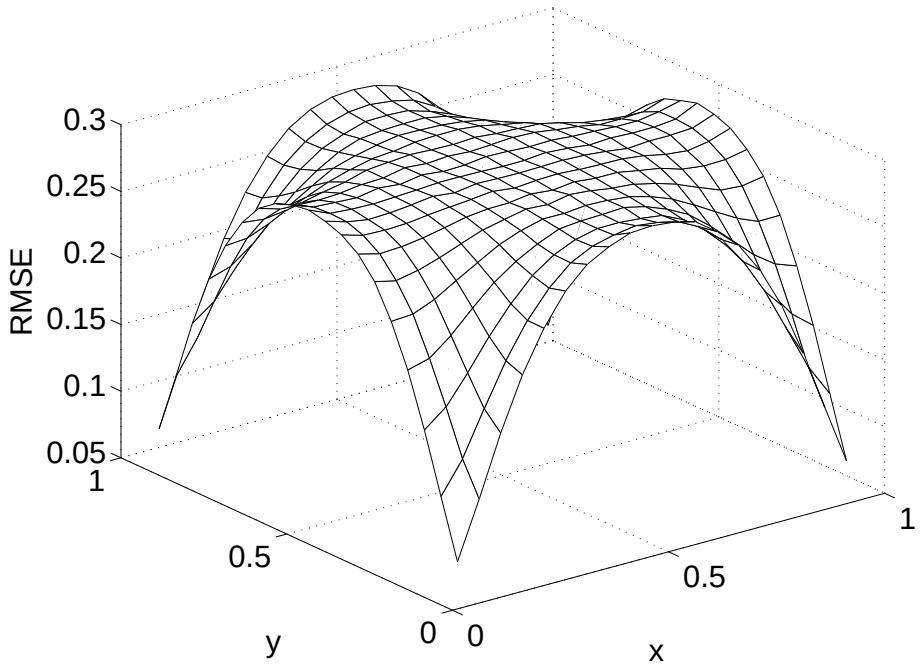

Figure 2.5: RMSE of the linear combination estimator versus the coordinates of one blindfolded node given $n_p = 3$ and $\sigma = 5$.

and the sample bias length is defined as

$$\text{(sample) bias length} \equiv \|B(\hat{\theta})\| = \left\| \frac{1}{N_{\text{sim}}} \sum_{i=1}^{N_{\text{sim}}} \hat{\theta}_i - \theta \right\|, \tag{2.72}$$

i.e., using sample mean to replace expectation.

The simulation results of estimation error under 10^5 trails with $n_p = 3$, $\sigma = 5$ in each observed location are provided in Figures 2.4–2.6 and in Table 2.3. All five algorithms have the similar RMSE shapes and the bounded LC has the smallest average, maximal, and minimal RMSEs. All these algorithms have the smallest RMSE near anchors; such a property is desirable since adding extra anchors presumably improves the estimation accuracy of the observed node. Although the bounded LC has the lowest RMSE, it has the highest length of bias. On the other hand, the MLE

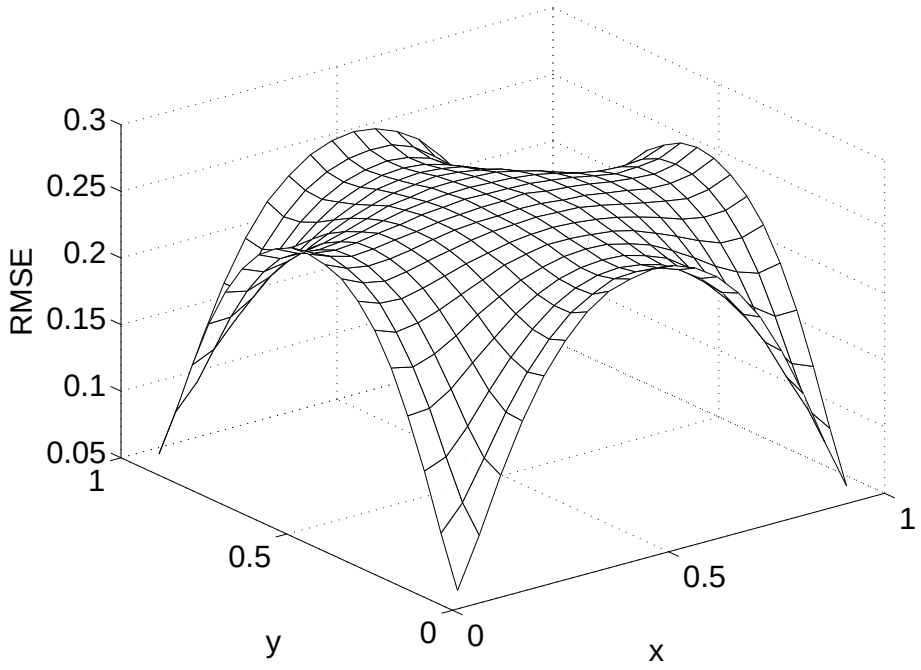

Figure 2.6: RMSE of the bounded linear combination estimator versus the coordinates of one blindfolded node given $n_p = 3$ and $\sigma = 5$.

has higher RMSEs than BLC: ML ranges, but it has the lowest bias length among all 5 different algorithms. The conventional LC has moderate RMSEs and bias.

The comparisons among these estimators for different n and σ with the same topology as in the above example are presented as follows. The simulation results with $n_p = 1.5$ and $\sigma = 5$ are shown in Table 2.4. First of all, the RMSE shapes still resembles a reversed bell. However, all estimators have worse accuracy than the case with $\sigma = 3$ because the range estimator's error is larger as shown in Figure 2.2. Opposite to the case with $\sigma = 3$, the avg. RMSE of the MLE is larger than that of the LC. The reason is that the LC using the UB ranges has better accuracy than the MLE using ML ranges especially when n_p is small as also seen in Figure 2.2. Similarly, the LC using ML ranges has much worse RMSEs than other methods.

Table 2.3: Extreme and Average Statistics of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for Various Estimators Given $n_p = 3$ and $\sigma = 5$

Method	Avg. RMSE	Max. RMSE	Position	Min. RMSE	Position	Avg. bias length
MLE	0.2346	0.2900	(.50, .50)	0.0591	(.95, .05)	0.0329
LC	0.2552	0.2902	(.05, .50)	0.0716	(.50, .95)	0.0368
BLC	0.2204	0.2569	(.55, .95)	0.0538	(.95, .05)	0.0472
LC: ML ranges	0.2876	0.3193	(.55, .70)	0.0848	(.95, .95)	0.0339
BLC: ML ranges	0.2271	0.2648	(.50, .55)	0.0580	(.95, .95)	0.0435

Table 2.4: Extreme and Average Statistics of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for Various Estimators Given $n_p = 1.5$ and $\sigma = 5$

Method	Avg. RMSE	Max. RMSE	Position	Min. RMSE	Position	Avg. bias length
MLE	0.4061	0.4624	(.50, .50)	0.1391	(.05, .05)	0.0864
LC	0.4043	0.4647	(.05, .50)	0.1453	(.05, .95)	0.0959
BLC	0.3700	0.4295	(.95, .50)	0.1400	(.95, .95)	0.1107
LC: ML ranges	0.4957	0.5712	(.50, .50)	0.1652	(.95, .95)	0.0668
BLC: ML ranges	0.3821	0.4429	(.50, .95)	0.1476	(.05, .95)	0.1144

Table 2.5 illustrates a case with much better SNR: $n_p = 5$ and $\sigma = 5$. In this case, the difference between estimators using the UB and ML range estimators is small and BLC still has the best RMSE.

In conclusion of various algorithms in this topology under different SNR, the BLC has the lowest RMSE, but its drawback is larger bias. The LC has the slightly larger RMSE and bias than the MLE.

Table 2.5: Extreme and Average Statistics of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for Various Estimators Given $n_p = 5$ and $\sigma = 5$

Method	Avg. RMSE	Max. RMSE	Position	Min. RMSE	Position	Avg. bias length
MLE	0.1529	0.1714	(.65, .30)	0.0502	(.05, .05)	0.0196
LC	0.1697	0.1861	(.05, .50)	0.0667	(.95, .95)	0.0214
BLC	0.1471	0.1614	(.30, .50)	0.0451	(.95, .95)	0.0248
LC: ML ranges	0.1817	0.1957	(.05, .45)	0.0740	(.95, .95)	0.0202
BLC: ML ranges	0.1502	0.1653	(.70, .35)	0.0473	(.95, .95)	0.0221

The lower bound for the LC estimator is derived with the known relative directions; the approximation and the upper bound for the LC estimator are given by assigning random directions. The upper bound and approximation are under the assumption using optimal weights $\mathbf{a}_{\text{opt}}$ and the upper bound using equal weights. As mentioned in Section 2.4, the iteratively found directions in the LC algorithm could be worse than random guess used in the upper bound if the SNR is extremely low. As investigated with fixed $\sigma = 5$, the approximation working as an upper bound in both $n = 3$ and $n = 5$ in Table 2.6 and Table 2.7, respectively. However, when $n = 1.5$ in Table 2.8, the RMSE of the approximation can be smaller than LC and and LC: $\mathbf{a}_{\text{opt}}$. More explicitly, it happens in 353 out of 361 locations in LC: $\mathbf{a}_{\text{opt}}$ and 239 out of 361 locations in LC. In the case of 4 corner anchors, the randomly guessed angles might be better when n_p/σ is lower than 2.7/5 according to the approximation. In other words, the approximation cannot be an upper bound in the low SNR regime.

Moreover, Tables 2.6–2.8 present the results of the LC estimators using equal weights, the upper bounds, and the lower bounds. The difference between LC with equal weight and LC in average and maximal RMSEs is not extremely big in high

Table 2.6: Extreme and Average Statistics of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for LC Estimators and Bounds Given $n_p = 3$ and $\sigma = 5$

Method	Avg. RMSE	Max. RMSE	Position	Min. RMSE	Position	Avg. bias length
Random guess	0.5571	0.7570	(.05, .05)	0.4091	(.50, .50)	0.3631
LC: upper bound	0.5721	0.8168	(.05, .05)	0.3806	(.50, .50)	0.3631
LC: equal weight	0.3282	0.3469	(.05, .50)	0.3054	(.95, .95)	0.0146
LC: approx.	0.3196	0.3806	(.50, .50)	0.0960	(.05, .05)	0.0696
LC: $\mathbf{a}_{\text{opt}}$	0.2812	0.3213	(.45, .60)	0.0887	(.95, .95)	0.0518
LC	0.2552	0.2902	(.05, .50)	0.0716	(.50, .95)	0.0368
LC: lower bound	0.1139	0.1408	(.50, .50)	0.0280	(.05, .05)	0

SNR. However, the minimal RMSE of LC with equal weight is much huger and its RMSE is almost the same for the blindfolded node at different locations as also shown in Figure 2.7. This is because the LC equal weight takes no advantage from closer anchors, which have more accurate location estimates. It also suggests that adding more anchors improves the RMSE less for the equal weight combination. The upper bounds by equal weight are loose and are worse than random guess in some cases. However, as the number of anchors increases, it RMSE can be lower than the RMSE of the randomly guessed estimates. The lower bound increases as the SNR decreases as expected. The gap between the lower bound and any LC estimator is smaller for larger SNR because the proposed iterative angle estimation is closer to true angle as the SNR increases.

2.5.2 Unit Circle

In the second topology, the blindfolded node is at the origin and anchors are uniformly placed inside a unit circle. Unlike the previous topology, the anchors'

Table 2.7: Extreme and Average Statistics of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for LC Estimators and Bounds Given $n_p = 5$ and $\sigma = 5$

Method	Avg. RMSE	Max. RMSE	Position	Min. RMSE	Position	Avg. bias length
LC: upper bound	0.5569	0.8022	(.05, .05)	0.3631	(.50, .50)	0.3631
LC: equal weight	0.1999	0.2170	(.95, .75)	0.1733	(.50 .50)	0.0088
LC: approx.	0.3060	0.3631	(.50, .50)	0.0933	(.05, .05)	0.0696
LC: $\mathbf{a}_{\text{opt}}$	0.1827	0.1943	(.80, .75)	0.0797	(.95, .95)	0.0292
LC	0.1697	0.1861	(.05, .50)	0.0667	(.95, .95)	0.0214
LC: lower bound	0.0667	0.0825	(.50, .50)	0.0164	(.05, .05)	0

Table 2.8: Extreme and Average Statistics of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for LC Estimators and Bounds Given $n_p = 1.5$ and $\sigma = 5$

Method	Avg. RMSE	Max. RMSE	Position	Min. RMSE	Position	Avg. bias length
LC: upper bound	0.6575	0.9015	(.05, .05)	0.4747	(.50, .50)	0.3631
LC: equal weight	0.6083	0.6456	(.50, .95)	0.5862	(.60, .45)	0.0557
LC: approx.	0.3934	0.4747	(.50, .50)	0.1113	(.05, .05)	0.0696
LC: $\mathbf{a}_{\text{opt}}$	0.4575	0.5873	(.45, .50)	0.1059	(.95, .95)	0.0665
LC	0.4043	0.4647	(.05, .50)	0.1453	(.05, .95)	0.0959
LC: lower bound	0.2560	0.3167	(.50, .50)	0.0629	(.05, .05)	0

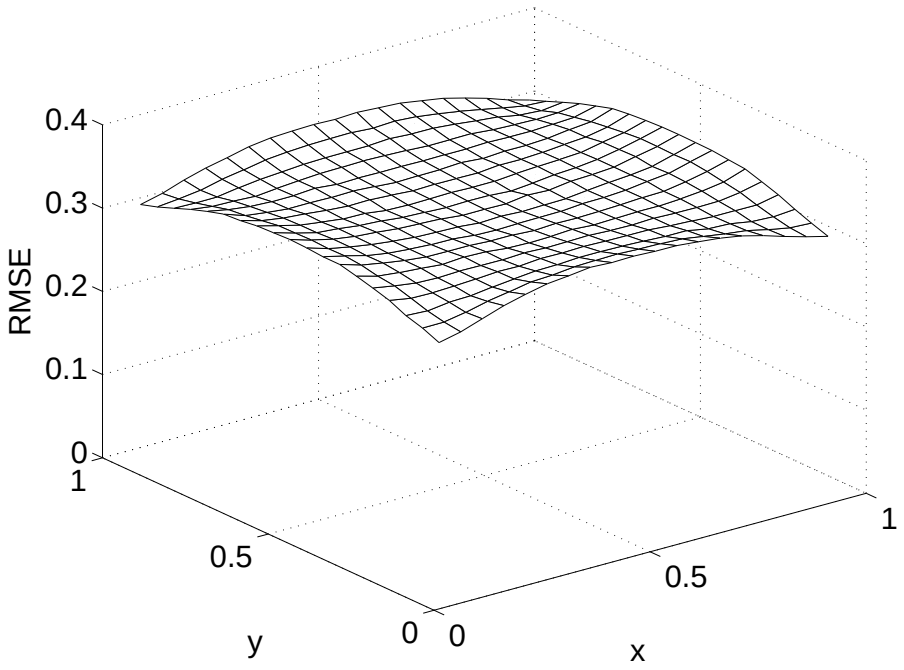

Figure 2.7: RMSE of the linear combination estimator with equal weight versus the coordinates of one blindfolded node given $n_p = 3$ and $\sigma = 5$.

locations now are random variables as well as the received power measurements. It now makes the distances and angles random variables as mentioned in Section 2.4. Then the MSE needs to average over $\hat{\theta}$, $D \equiv [D_1, \ldots, D_n]$, and $\Phi \equiv [\Phi_1, \ldots, \Phi_n]$, i.e., $\text{MSE}(\hat{\theta}) = \mathbb{E}_{\hat{\theta}D\Phi}[\|\hat{\theta} - \theta\|^2]$. However, in simulation, each anchor's placement only runs one set of random power measurements. The sample RMSE uses the same format as in (2.70) and the average over $\hat{\theta}$ is actually from different topologies. In simulation, the number of trails is 10^5, the stopping ϵ is 10^{-6}, and the permitted maximum number of iterations is 10^4 for all LCs. For the MLE, the search space is implemented in the polar coordinate (r, φ) instead of the Cartesian coordinate (x, y). The conversion between two coordinate systems is $(x, y) = (r\cos\varphi, r\sin\varphi)$. The search space for r is denoted by $\mathcal{S}_r = \{\Delta, \ldots, 1-\Delta, 1\}$ and the number of candidates

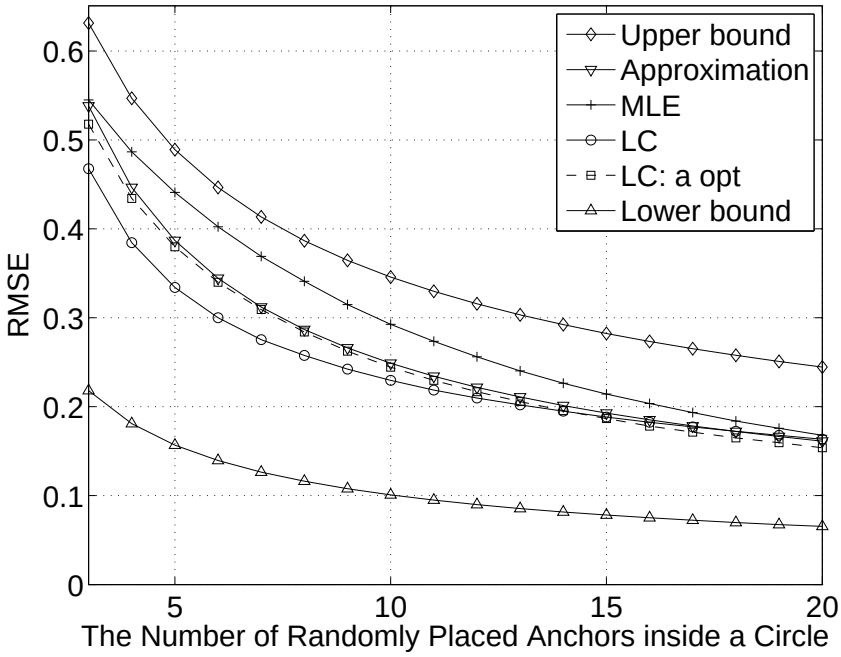

Figure 2.8: RMSE of various algorithms versus the number of randomly placed anchors inside a unit circle with $n_p = 2$ and $\sigma = 5$.

is $|\mathcal{S}_r| = 1/\Delta$ where Δ is the step size for r. The step size for φ is $\Delta_\varphi = 2\pi/|\mathcal{S}_r|$ which results in the search space for angle $\mathcal{S}_\varphi = \{0, \Delta_\varphi, \ldots, 2\pi - \Delta_\varphi\}$ and $|\mathcal{S}_\varphi| = |\mathcal{S}_r|$. Finally, the search space in the circle case is given by $\mathcal{S} = \{\mathcal{S}_r \times \mathcal{S}_\varphi\} \cup \{0\}$. In the simulations, Δ is set to be 0.005 which makes $|\mathcal{S}_r| = |\mathcal{S}_\varphi| = 200$ and $\Delta_\varphi = \pi/100$. Thus there are $40,001$ candidates.

The sample RMSEs versus the number of randomly placed anchors are shown in Figures 2.8–2.10 with $n = 2$, 3, and 5, respectively. First of all, these plots illustrate how the RMSE decreases when the number of anchors increases. In this topology, because $\mathbb{E}_D[D_k^2] = r^2/2 = 1/2$ where r is the radius of a circle, the upper bound

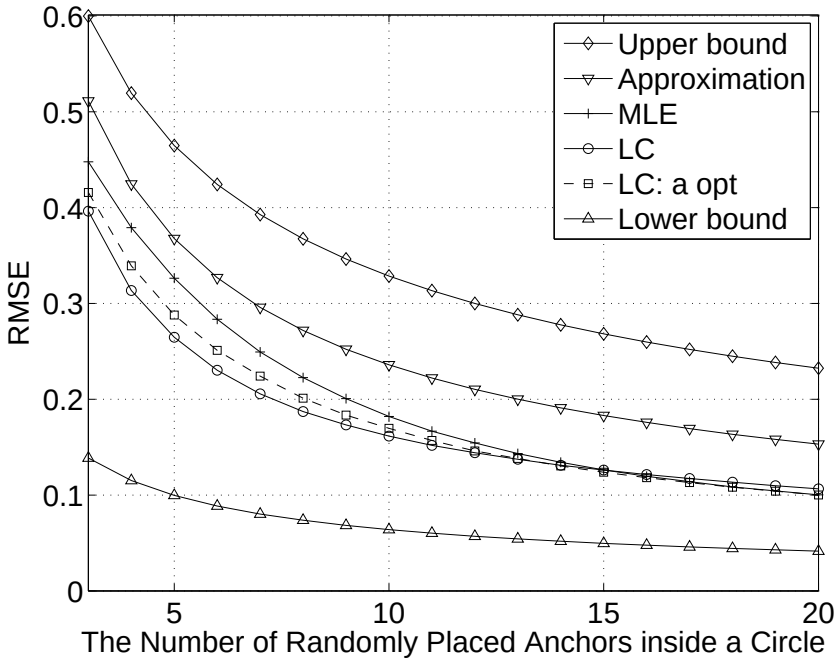

Figure 2.9: RMSE of various algorithms versus the number of randomly placed anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

with equal weight can be simplified as

$$\mathrm{MSE}(\hat{\theta}) \leq \frac{\exp(b^2) + 1}{2n}. \tag{2.73}$$

Although a simple computation for the above equation, simulations show a huge gap between the upper bound and the approximation. For the target RMSE is 0.2, MLE requires 9 anchors and LC requires 7 anchors under $n_p/\sigma = 3/5$; for target RMSE is 0.1, MLE and all LC variations need about 20 anchors under $n_p/\sigma = 3/5$. It is also found that the plots of all LC algorithms is closer to the lower bound when the SNR n_p/σ increases because the angle estimation is better. On the other hand, RMSEs are closer to the upper bound when SNR is lower and SNR $= 2/5$ seems to be the limit of approximation to be viewed as an upper bound in the circle topology.

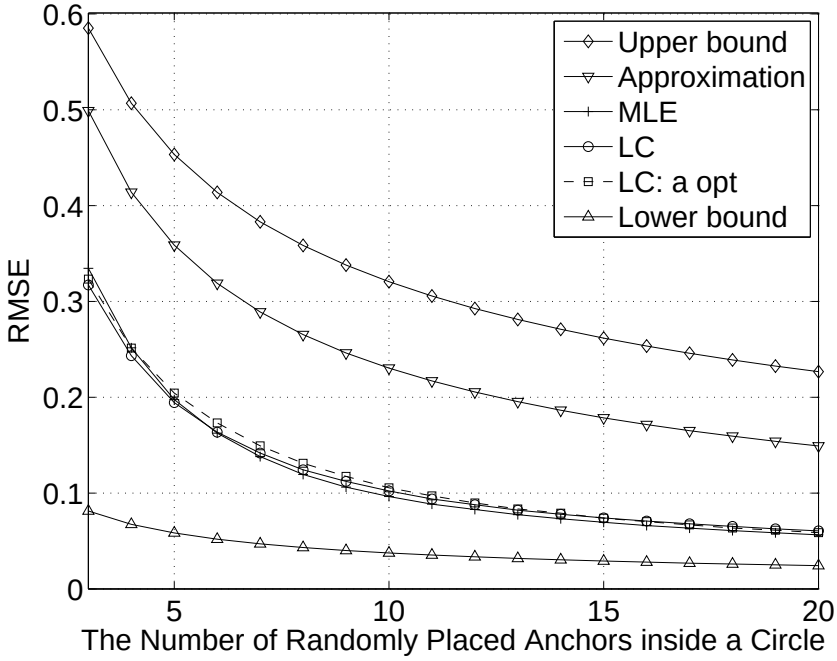

Figure 2.10: RMSE of various algorithms versus the number of randomly placed anchors inside a unit circle with $n_p = 5$ and $\sigma = 5$.

Moreover, the MLE works worse than all LCs under this topology in most cases, especially when the SNR or the density of anchors is low. LC performs better than LC: $\mathbf{a}_{\text{opt}}$ when few anchors exist or SNRs are low. Again, LC: $\mathbf{a}_{\text{opt}}$ is guaranteed to have the lowest MSE if the true directions are given. On the other hand, LC: $\mathbf{a}_{\text{opt}}$ is better in other situations. In general, the difference between LC and LC: $\mathbf{a}_{\text{opt}}$ is small. Thus instead of the loose upper bound by equal weight, the approximation using optimal weights can clip the accuracy of LC in most situations.

2.6 Computation and Communication Costs

This section discusses computation and communications costs for both the LC family and the MLE. Because the number of iterations is the key for the above two costs in iterative methods, several behaviors of iterations in LC are presented.

Table 2.9: Average Number of Iterations for Various LC Estimator as One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ Given $\sigma = 5$

Method	$n_p = 1.5$	$n_p = 3$	$n_p = 5$
LC	33.00	25.39	37.52
BLC	23.37	17.14	28.18
LC: ML ranges	39.46	40.17	41.28
BLC: ML ranges	25.37	21.26	29.42

The time complexity for the non-cooperative linear combination estimator is $O(nK)$ where n is the number of anchors and K is the number of iterations [45]. K is crucial to the time complexity in LC. Table 2.9 provides the average the number of iterations required for various LCs with $\epsilon = 10^{-8}$ in 4-anchor cases under different SNRs. All LCs with $n = 5$ have the most iterations and most of LCs with $n = 3$ give the fewest iterations. The BLC requires fewest iterations and the LC using ML ranges needs largest iterations. Besides the average number of iterations, Figures 2.11 and 2.11 illustrate distributions of the number iterations at positions $(0.05, 0.05)$ and $(0.50, 0.50)$, respectively. When a blindfolded is near an anchor, the LC estimator requires more iterations to converge. On the other hand, the number of iterations are much fewer and 80 percent of the cases require less than 20 iterations. Moreover, $\epsilon = 10^{-8}$ is strict, in which the change of RMSE is ignorable after smaller number of iterations. Clearly, smaller ϵ results in fewer iterations and it does not affect the RMSE significantly.

Figures 2.13–2.15 present the RMSE versus the number of iterations given randomly placed anchors inside a unit circle. The estimates for the LC estimator are stable after 100 iterations on average in all three different n_p. Moreover, the time complexity of the LC estimator related to the number of iterations varies with differ-

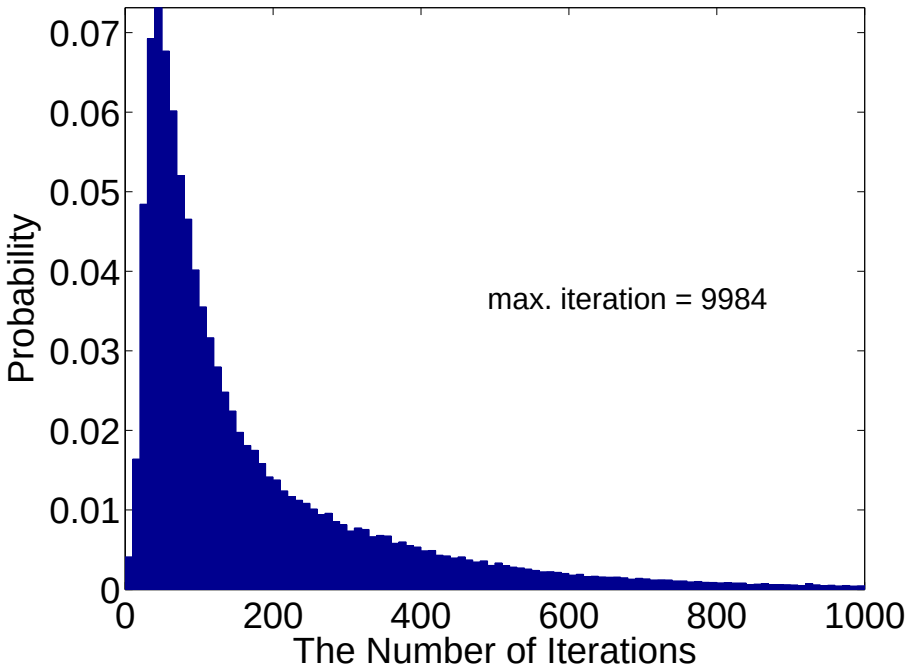

Figure 2.11: Distribution of iterations for the linear combination estimator at position $(0.05, 0.05)$ given $n_p = 3$ and $\sigma = 5$.

ent channel parameters. The lowest RMSE also happens before the RMSE becomes stable, suggesting that if the algorithm stops earlier, it may perform better in situations with lower SNR and fewer anchors. The accuracy of $n_p = 1.5$ with few anchors is even better without any iterations. It shows that the angle estimation of the LC algorithm performs worse than a random angle guess in the low SNR regime again.

For the MLE using an exhaustive search, finding the minimum in a search space is the key for time complexity. Unlike the LC estimator, the MLE with exhaustive search has the same running time under different channel parameters if the search space remains unchanged. Since searching for a minimum is linear in time, the time complexity for MLE is linear in both the number of anchors and the size of search space, i.e., $O(n|\mathcal{S}|)$. Because the searching size is usually much larger than the num-

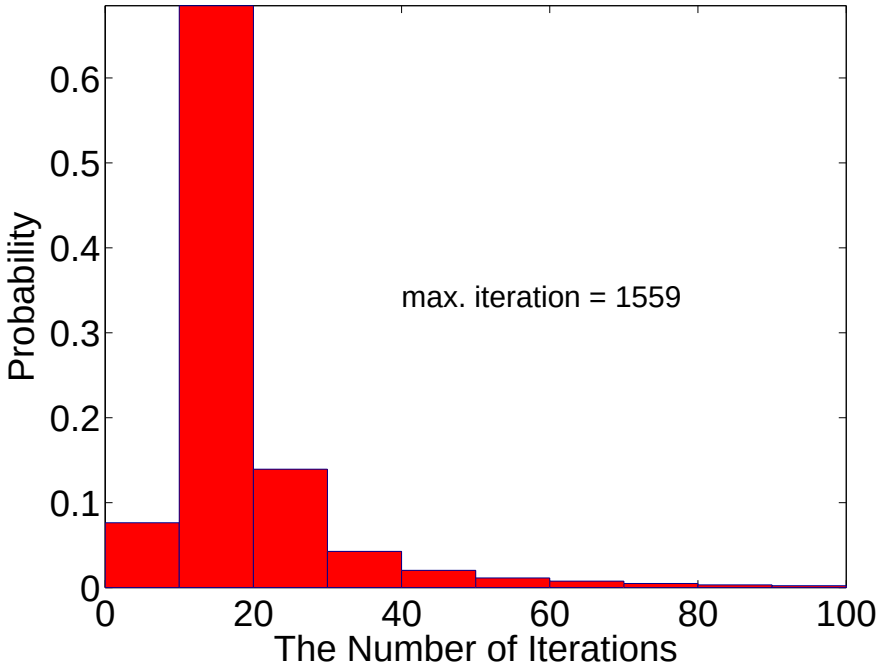

Figure 2.12: Distribution of iterations for the linear combination estimator at position $(0.50, 0.50)$ given $n_p = 3$ and $\sigma = 5$.

ber of anchors, the computational time is dominated by the size $|\mathcal{S}|$. Understanding how the size affects RMSE is essential for an exhaustive search. Mimicking the definition of relative error, the relative difference of RMSE is defined to quantify the accuracy gap between two step sizes:

$$\delta\text{RMSE} \equiv \frac{\text{RMSE} - \text{RMSE}_0}{\text{RMSE}_0} \tag{2.74}$$

where RMSE_0 is the reference RMSE with a very small step size. Figure 2.16 and Table 2.10 present the relative RMSE as it depends on step size given $\text{RMSE}_0 = 0.002$. Using a step size smaller than 0.01 (10,001 candidates) should be good enough because the absolute value of relative RMSE is less than 2 percent.

The running time of both LC and MLE are calculated by the function `cputime`

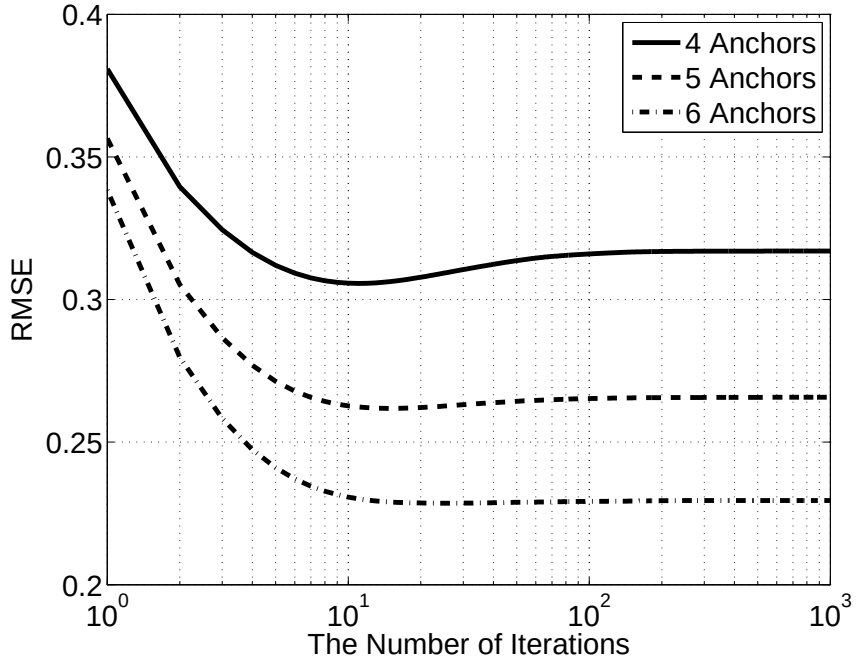

Figure 2.13: RMSE of LC versus the number of iterations for randomly placed anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

Table 2.10: Related Statistics of Non-cooperative MLE using Various Step Sizes for Randomly Placed Anchors Inside a Unit Circle with $n_p = 3$ and $\sigma = 5$.

Step size	# of Candidates	Max. δRMSE	Min. δRMSE
0.1	101	0.1163	-0.0693
0.05	401	0.0250	-0.0228
0.01	10,001	0.0025	-0.0128
0.005	40,001	0.0046	-0.0058
0.002	250,001	0	0

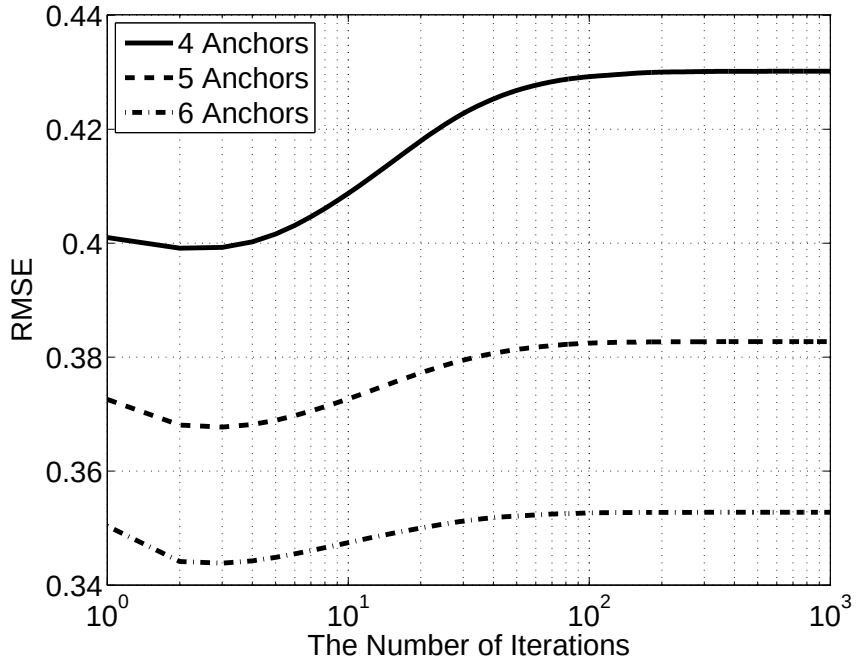

Figure 2.14: RMSE of LC versus the number of iterations for randomly placed anchors inside a unit circle with $n_p = 1.5$ and $\sigma = 5$.

of Matlab. Matlab (R2007b) is installed in a personal computer equipped with an Intel Xeon 2.66 GHz CPU and 4 GB RAM; the OS is Linux version 2.6.26-2-amd64 (Debian). The results are presented in Figure 2.17 under 10^3 trails, showing that the MLE with a exhaustive search only can execute faster than the LC estimator with at most 8 anchors and a step size 0.05 (401 candidates). It also demonstrates that the MLE is linear with the number of anchors. On the other hand, the running time of LC decreases and stabilizes when there are more anchors, because the number of iterations decreases given more anchors as shown in Figure 2.18.

Non-cooperative localization requires only one-time transmission from each anchor to a single blindfolded node and no information exchange betweens nodes. The location of the anchor is the only necessary information. Both the LC estimator

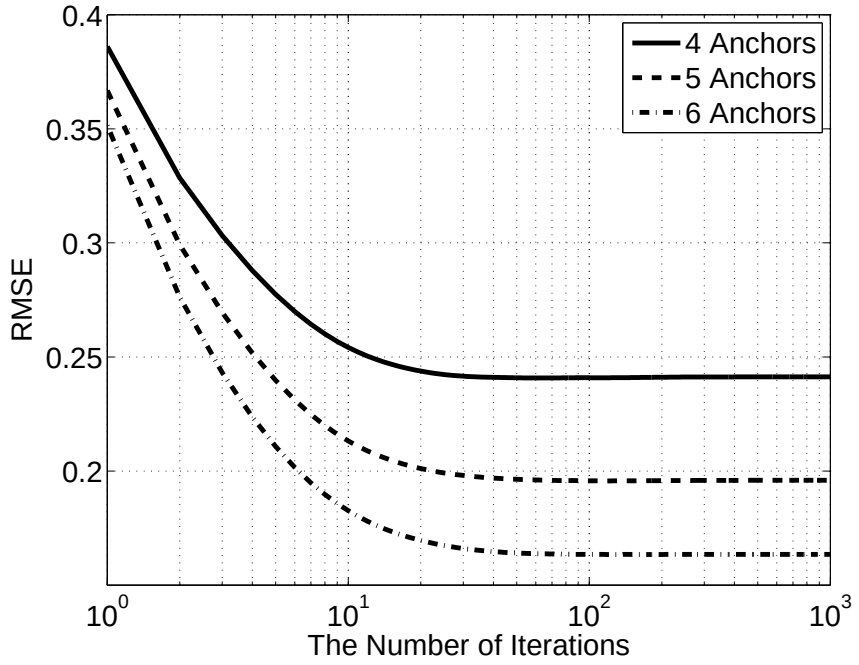

Figure 2.15: RMSE of LC versus the number of iterations for randomly placed anchors inside a unit circle with $n_p = 5$ and $\sigma = 5$.

and MLE demand no more communications during iterations or the search for the minimum. Thus the communication cost is the total bits that convey all anchors' locations.

2.7 Convergence Analysis

Checking the convergence of an iterative algorithm reveals when and how fast it stops. Let $\mathbf{T}$ be an iterative function and the iterations can be written as

$$\hat{\theta}^{(k)} = \mathbf{T}(\hat{\theta}^{(k-1)}) = \mathbf{T}^{(2)}(\hat{\theta}^{(k-2)}) = \cdots = \mathbf{T}^{(k)}(\hat{\theta}^{(0)}) \tag{2.75}$$

where $\mathbf{T}^{(k)}$ represents to perform iteration k times, $\hat{\theta}^{(k-1)}$ is the estimates after $k-1$ iterations, and $\hat{\theta}^{(0)}$ is the initial guess. In the non-cooperative LC estimator, the

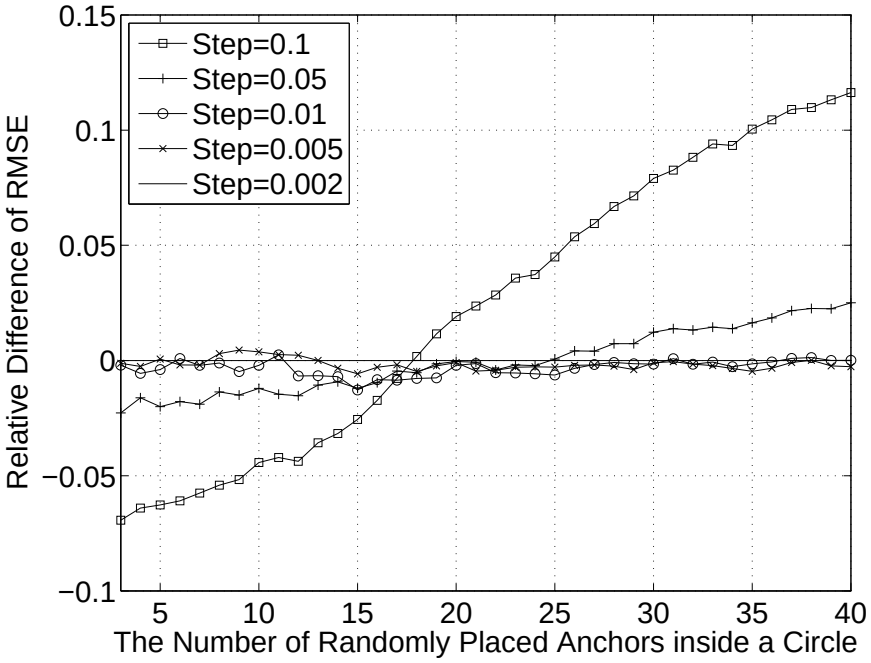

Figure 2.16: The relative difference of RMSE where RMSE_0 has the step size 0.002 for randomly placed anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

iterative function is denoted by

$$\mathbf{T}(\hat{\theta}^{(k)}) \equiv \sum_{i=1}^{n} a_i \left([x_i, y_i] + \hat{d}_i \frac{[\hat{x}^{(k)} - x_i, \hat{y}^{(k)} - y_i]}{\|[\hat{x}^{(k)} - x_i, \hat{y}^{(k)} - y_i]\|} \right) = \hat{\theta}^{(k+1)}. \tag{2.76}$$

Equivalently,

$$f(\hat{x}^{(k)}, \hat{y}^{(k)}) \equiv \sum_{i=1}^{n} a_i x_i + \sum_{i=1}^{n} a_i \hat{d}_i \frac{\hat{x}^{(k)} - x_i}{\sqrt{(\hat{x}^{(k)} - x_i)^2 + (\hat{y}^{(k)} - y_i)^2}} = \hat{x}^{(k+1)}, \tag{2.77}$$

$$g(\hat{x}^{(k)}, \hat{y}^{(k)}) \equiv \sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i \hat{d}_i \frac{\hat{y}^{(k)} - y_i}{\sqrt{(\hat{x}^{(k)} - x_i)^2 + (\hat{y}^{(k)} - y_i)^2}} = \hat{y}^{(k+1)}. \tag{2.78}$$

The iterative method $\mathbf{T}$ is said convergent to θ^* if for each $\epsilon > 0$, there exists an iterated step K such that $k > K$ implies $|\hat{\theta}^{(k)} - \theta^*| < \epsilon$. It is called *local convergence*

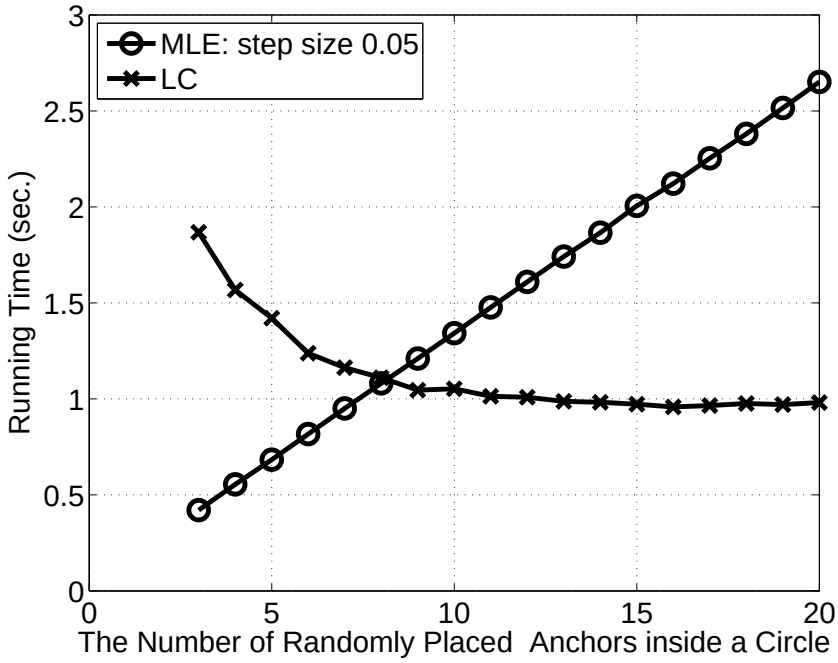

Figure 2.17: Computation time of LC and MLE with search space for randomly placed anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$ under 10^3 trails.

if the method converges θ^* given the initial guess $\hat{\theta}^{(0)}$ is sufficiently close to θ^*. On the other hand, *global convergence* means any initial guess will converge to θ^*. The rate of convergence classifies iterative algorithms and tells how fast they converge [46].

To ensure the convergence, θ^* needs to first exist and secondly be achievable. These two issues are discussed in the following two subsections: fixed points and stability, respectively.

2.7.1 Fixed Points

In an iterative function $\mathbf{T}$, if $\theta^* = \mathbf{T}(\theta^*)$, θ^* is called a *fixed point* or *equilibrium* [46, 47]. Finding a fixed point by directly solving $\theta^* = \mathbf{T}(\theta^*)$ may not be easy. An iterative algorithm could solve the above equation by sequentially approaching the fixed point.

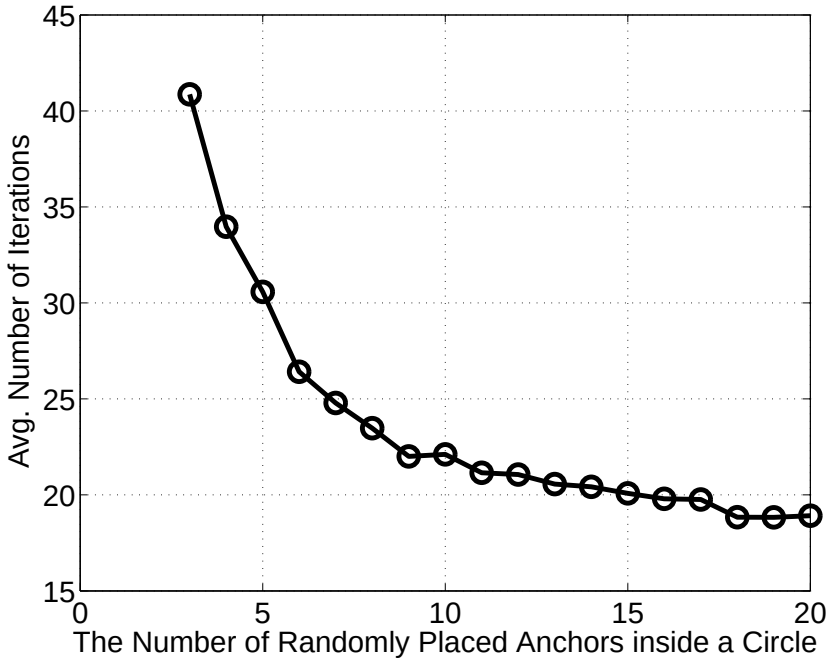

Figure 2.18: Average number of iterations of LC versus the number of randomly placed anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$ under 10^3 trails.

There are several famous sufficient conditions to ensure the existence of a fixed point. Banach's contraction mapping principle [48] provides not only the existence but also the uniqueness. Moreover, any initial point converges to this unique point. Roughly speaking, the condition requires the distance between two arbitrary points getting closer after each iteration. Unfortunately, the non-cooperative LC estimator fails to satisfy this contraction especially when the two points are too far or too close. In fact, the LC estimator may have multiple fixed points.

The Brouwer fixed point theorem [49] guarantees the existence of fixed points for a continuous function in a nonempty, compact, and convex set. The estimate result of the LC estimator can be bounded by a compact and convex set $[\min\{x_i - d_i\}, \max\{x_i + d_i\}] \times [\min\{y_i - d_i\}, \max\{y_i + d_i\}]$. However, the LC estimator is

not continuous when $\hat{\theta}^{(k)}$ is at any anchor's location $[x_i, y_i]$ because the direction $\mathbf{v}_i^{(k)} = \mathbf{0}$. Thus the LC estimator does not satisfy this theorem. Herings et al. provide a fixed-point theorem for discontinuous functions [50]. The condition is called the locally gross direction preserving property: for any non-fixed point and any two of its neighbor points z_1 and z_2, if the inner product of vector $\mathbf{T}(z_1) - z_1$ and $\mathbf{T}(z_2) - z_2$ are non-negative, a fixed point exists. For the LC, the only non-fixed points which should be checked are the anchors' locations. Unfortunately, the LC violates this condition because sine and cosine can be any numbers from -1 to 1 near the anchors while estimating angle $\mathbf{v}_i^{(k)}$ in 2.35.

2.7.2 Stability

In this subsection, an iterative algorithm is assumed to have at least one fixed point. However, the existence of fixed points is not guaranteed converge of the algorithm. Thus it is essential to ascertain the achievability. A fixed point is called *asymptotically stable* if all neighbor points converge to it [51]. In contrast, if the fixed point is *unstable*, all its neighbors will diverge from it. To achieve the unstable fixed point, the algorithm must start exactly at the fixed point. The *stable* property only guarantees that the nearby solutions remains near to the fixed point.

Consider the 2-D case and assume $\theta^* = [x^* \; y^*]$ is the fixed point i.e., $\theta^* = \mathbf{T}(\theta^*)$. The fixed point is asymptotically stable if and only if the spectral radius of its Jacobian matrix is less than 1 [51]. To simplify the notation, let $\theta^* = \theta = [x \; y]$. Then the Jacobian matrix $\mathbf{T}(\theta)$ for the fixed point is

$$\mathbf{T}'(\theta) = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} & \frac{\partial f(x,y)}{\partial y} \\ \frac{\partial g(x,y)}{\partial x} & \frac{\partial g(x,y)}{\partial y} \end{bmatrix} \tag{2.79}$$

where

$$\begin{aligned}
\frac{\partial f(x,y)}{\partial x} &= \sum_{i=1}^{n} a_i \hat{d}_i \frac{1}{\sqrt{(x-x_i)^2+(y-y_i)^2}} \left[1 - \frac{(x-x_i)^2}{(x-x_i)^2+(y-y_i)^2}\right] \\
&= \sum_{i=1}^{n} a_i \hat{d}_i \frac{(y-y_i)^2}{[(x-x_i)^2+(y-y_i)^2]^{3/2}} \qquad (2.80) \\
\frac{\partial g(x,y)}{\partial y} &= \sum_{i=1}^{n} a_i \hat{d}_i \frac{(x-x_i)^2}{[(x-x_i)^2+(y-y_i)^2]^{3/2}} \qquad (2.81) \\
\frac{\partial f(x,y)}{\partial y} &= -\sum_{i=1}^{n} a_i \hat{d}_i \frac{(x-x_i)(y-y_i)}{[(x-x_i)^2+(y-y_i)^2]^{3/2}} = \frac{\partial g(x,y)}{\partial x} \qquad (2.82)
\end{aligned}$$

when θ^* is not equal to the locations of any anchors. Moreover, the spectral radius of $\mathbf{T}'(\theta)$ is defined as

$$\rho(\mathbf{T}'(\theta)) \equiv \max(\{|\lambda_1|, |\lambda_2|\}) \qquad (2.83)$$

where λ_1 and λ_2 are eigenvalues of $\mathbf{T}'(\theta)$. On the other hand, they are solutions of $\lambda^2 - (\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y})\lambda + \frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial f}{\partial y}\frac{\partial g}{\partial x} = 0$. Thus

$$\begin{aligned}
\lambda &= \frac{(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}) + \sqrt{(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y})^2 - 4(\frac{\partial f}{\partial x}\frac{\partial g}{\partial y} - \frac{\partial f}{\partial y}\frac{\partial g}{\partial x})}}{2} \\
&= \frac{(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}) + \sqrt{(\frac{\partial f}{\partial x} - \frac{\partial g}{\partial y})^2 + 4\frac{\partial f}{\partial y}\frac{\partial g}{\partial x}}}{2}. \qquad (2.84)
\end{aligned}$$

By the Cauchy-Schwarz inequality,

$$\begin{aligned}
&\left(\sum_{i=1}^{n} a_i \hat{d}_i \frac{(y-y_i)^2}{[(x-x_i)^2+(y-y_i)^2]^{3/2}}\right)\left(\sum_{i=1}^{n} a_i \hat{d}_i \frac{(x-x_i)^2}{[(x-x_i)^2+(y-y_i)^2]^{3/2}}\right) \\
&\qquad \geq \left(\sum_{i=1}^{n} a_i \hat{d}_i \frac{(x-x_i)(y-y_i)}{[(x-x_i)^2+(y-y_i)^2]^{3/2}}\right)^2. \qquad (2.85)
\end{aligned}$$

Equivalently, $\frac{\partial f}{\partial x}\frac{\partial g}{\partial y} \geq \frac{\partial f}{\partial y}\frac{\partial g}{\partial x}$ and it implies that $(\frac{\partial f}{\partial x} + \frac{\partial g}{\partial y})^2 \geq (\frac{\partial f}{\partial x} - \frac{\partial g}{\partial y})^2 + 4\frac{\partial f}{\partial y}\frac{\partial g}{\partial x}$ which

makes both eigenvalues larger than zero.

Therefore to know the convergence of the non-cooperative localization algorithm, it only needs to check if $\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}+\sqrt{(\frac{\partial f}{\partial x}-\frac{\partial g}{\partial y})^2+4\frac{\partial f}{\partial y}\frac{\partial g}{\partial x}}<2$ given a fixed point (x,y). That is,

$$\frac{\partial f}{\partial x}+\frac{\partial g}{\partial y}-1>\frac{\partial f}{\partial x}\frac{\partial g}{\partial y}-\frac{\partial f}{\partial y}\frac{\partial g}{\partial x} \tag{2.86}$$

to ensure the convergence. Moreover, a smaller spectral radius makes the iterative method converge faster. One stronger condition is

$$\sum_{i=1}^{n}\frac{a_i\hat{d}_i}{\sqrt{(x-x_i)^2+(y-y_i)^2}}<1 \tag{2.87}$$

Figure 2.19 illustrates an example of the average spectral radius (ρ) of a blindfolded node inside a unit square given $n_p=3$ and $\sigma=5$. The above simulations have 10^5 trails and the settings are the same as the one in Section 2.5 except the algorithm now stops when both $\epsilon<10^{-8}$ and $\rho<1$, unless the number of iterations are more than 10^4. The MSEs with and without this modification are almost identical. More importantly, this modification ensures that all spectral radii are less than 1, i.e., it makes all fixed points stable. It is found that the spectral radius is small when the blindfolded node is at the centre of the unit square and becomes larger as the node moves closer to the corner. The larger spectral radius coincides with the more required iterations when a blindfolded is near an anchor. The extreme and average spectral radius with different channel exponents n_p are presented in Table 2.11. The table shows that spectral radius becomes larger when the n_p is bigger or in the higher SNR regime.

Therefore, a solution of an iterative algorithm requires finding a fixed point and then checking its spectral radius to ensure stability. To conclude this section, it is

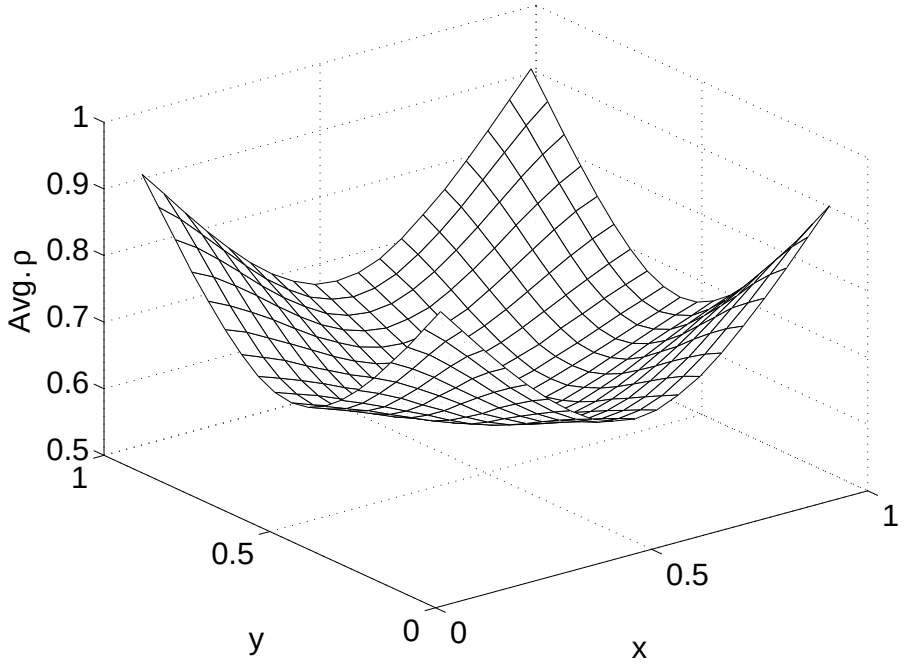

Figure 2.19: Average spectral radius of the linear combination estimator versus the coordinates of one blindfolded node given $n_p = 3$ and $\sigma = 5$.

Table 2.11: Extreme and Average Spectral Radius of One Single Blindfolded Node at $\{0.05, 0.1, \ldots, 0.9, 0.95\} \times \{0.05, 0.1, \ldots, 0.9, 0.95\}$ for LC Estimators Given and $\sigma = 5$

n_p	Avg. ρ	Max. ρ	Position	Min. ρ	Position
1.5	0.5859	0.7527	(0.95, 0.05)	0.5529	(0.55, 0.55)
3	0.7031	0.9254	(0.95, 0.05)	0.5992	(0.50, 0.50)
5	0.7464	0.9656	(0.95, 0.05)	0.5659	(0.50, 0.50)

important to return to the LC algorithm. To the best of my knowledge about mathematical theorem, the LC method is not guaranteed to converge. However, those theorems are sufficient conditions and the LC may still converge. Thus, the convergence is investigated by simulations. All previous simulations by the LC estimator have at least one fixed point and the corresponding spectral radius is smaller than 1. The non-cooperative LC algorithm thus converges in all trails in this dissertation. Moreover, the different initial points within an area end up in near fixed points. The convergence rates are slow when the blindfolded node is near an anchor. However, changing an initial point can make convergence faster.

2.8 Estimation under Unknown Propagation Model Parameters

In the previous discussions, the propagation model parameters, n_p, σ and $p_0(d_0)$ are all assumed to be perfectly known. However, these parameters may not be revealed in some applications. This section estimates these parameters and investigates the performance of localization using estimated parameters.

2.8.1 Estimation of $p_0(d_0)$ and n_p

This subsection discusses estimation of n_p and $p_0(d_0)$ because they are essential for the MLE and the LC estimator based on ML ranges. The received power is governed by a log-normal distribution as shown in (2.3):

$$p_{\mathrm{dB}} = p_{0,\mathrm{dB}} - 10n_p \log_{10}\left(\frac{d}{d_0}\right) + Z. \tag{2.88}$$

Because p_{dB} is linear with $\log_{10}\left(\frac{d}{d_0}\right)$, it is desired to find a straight line with slope $-\hat{n}_p$ and intercept $\hat{p}_{0,\mathrm{dB}}$ to approximate the above relationship. The famous "least square approximation" [52, 41] is considered and the approximated line $\hat{p}_{0,\mathrm{dB}} - 10\hat{n}_p \log_{10}\left(\frac{d}{d_0}\right)$ minimizes the sum of the vertical distance squared from all observed

p_{dB}'s to the line.

Explicitly, the distances are assumed to be known as well as the corresponding received powers. These pairs are denoted by $[d_1\ p_{1,\mathrm{dB}}], \ldots, [d_q\ p_{q,\mathrm{dB}}]$. The least square method minimizes

$$J(\hat{p}_{0,\mathrm{dB}},\ \hat{n}_p) = \sum_{i=1}^{q} \left(p_{i,\mathrm{dB}} - \hat{p}_{0,\mathrm{dB}} + 10\hat{n}_p \log_{10}\left(\frac{d_i}{d_0}\right)\right)^2. \tag{2.89}$$

Then the estimates are obtained by solving

$$\frac{\partial J}{\partial \hat{p}_{0,\mathrm{dB}}} = 0 \ \text{ and } \ \frac{\partial J}{\partial \hat{n}_p} = 0. \tag{2.90}$$

The final solution for the least square method is given by

$$\hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{p}_{0,\mathrm{dB}} \\ \hat{n}_p \end{bmatrix} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{p} \tag{2.91}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & -10\log_{10}(d_1/d_0) \\ 1 & -10\log_{10}(d_2/d_0) \\ \vdots & \vdots \\ 1 & -10\log_{10}(d_q/d_0) \end{bmatrix} \tag{2.92}$$

and

$$\mathbf{p} = \begin{bmatrix} p_{1,\mathrm{dB}} \\ p_{2,\mathrm{dB}} \\ \vdots \\ p_{q,\mathrm{dB}} \end{bmatrix}. \tag{2.93}$$

On the order hand, the log likelihood function is given by

$$\mathcal{L}(p_{0,\mathrm{dB}},\ n_p) = -\frac{q}{2}\ln 2\pi - \frac{q}{2}\ln\sigma^2 - \frac{\sum_{i=1}^{q}\left(p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10n_p\log_{10}\left(\frac{d_i}{d_0}\right)\right)^2}{2\sigma^2}. \quad (2.94)$$

The maximum likelihood estimator is achieved by minimizing (2.89) which is exactly the same as the least square method.

To obtain the CRB for vector $[\hat{p}_{0,\mathrm{dB}}\ \hat{n}_p]^T$, several partial derivatives are needed:

$$\frac{\partial\mathcal{L}}{\partial p_{0,\mathrm{dB}}} = \frac{\sum_{i=1}^{q} p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10n_p\log_{10}\left(\frac{d_i}{d_0}\right)}{\sigma^2} \quad (2.95)$$

$$\frac{\partial\mathcal{L}}{\partial n_p} = -\frac{\sum_{i=1}^{q}\left(p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10n_p\log_{10}\left(\frac{d_i}{d_0}\right)\right)10\log_{10}\left(\frac{d_i}{d_0}\right)}{\sigma^2} \quad (2.96)$$

$$\frac{\partial^2\mathcal{L}}{\partial p_{0,\mathrm{dB}}^2} = -\frac{q}{\sigma^2} \quad (2.97)$$

$$\frac{\partial^2\mathcal{L}}{\partial n_p^2} = -\frac{\sum_{i=1}^{q} 100\log_{10}^2\left(\frac{d_i}{d_0}\right)}{\sigma^2} \quad (2.98)$$

$$\frac{\partial^2\mathcal{L}}{\partial n_p\partial p_{0,\mathrm{dB}}} = \frac{\sum_{i=1}^{q} 10\log_{10}\left(\frac{d_i}{d_0}\right)}{\sigma^2} = \frac{\partial^2\mathcal{L}}{\partial p_{0,\mathrm{dB}}\partial n_p} \quad (2.99)$$

The Fisher information matrix is

$$\begin{aligned}\mathbf{I}_2(\boldsymbol{\theta}) &= \begin{bmatrix} -\mathbb{E}\left[\frac{\partial^2\mathcal{L}}{\partial p_{0,\mathrm{dB}}^2}\right] & -\mathbb{E}\left[\frac{\partial^2\mathcal{L}}{\partial p_{0,\mathrm{dB}}\partial n_p}\right] \\ -\mathbb{E}\left[\frac{\partial^2\mathcal{L}}{\partial n_p\partial p_{0,\mathrm{dB}}}\right] & -\mathbb{E}\left[\frac{\partial^2\mathcal{L}}{\partial n_p^2}\right] \end{bmatrix} \\ &= \frac{1}{\sigma^2}\begin{bmatrix} q & -\sum_{i=1}^{q} 10\log_{10}\left(\frac{d_i}{d_0}\right) \\ -\sum_{i=1}^{q} 10\log_{10}\left(\frac{d_i}{d_0}\right) & 100\sum_{i=1}^{q}\log_{10}^2\left(\frac{d_i}{d_0}\right) \end{bmatrix}\end{aligned} \quad (2.100)$$

Then the CRB for $\hat{p}_{0,\mathrm{dB}}$ and $\hat{n}_p$ are $[\mathbf{I}_2^{-1}(\boldsymbol{\theta})]_{1,1}$ and $[\mathbf{I}_2^{-1}(\boldsymbol{\theta})]_{2,2}$, respectively.

In general, the above structure can be written as

$$\mathbf{p} = \mathbf{A}\boldsymbol{\theta} + \mathbf{Z} \tag{2.101}$$

which is called the linear model [41]. The resulting least square estimator and MLE as shown earlier is

$$\hat{\boldsymbol{\theta}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{p}. \tag{2.102}$$

and the Fisher information matrix is

$$\mathbf{I}(\boldsymbol{\theta}) = \frac{\mathbf{A}^T\mathbf{A}}{\sigma^2} \tag{2.103}$$

Moreover,

$$\begin{aligned} \frac{\partial L}{\partial \boldsymbol{\theta}} &= \frac{\mathbf{A}^T\mathbf{p} - \mathbf{A}^T\mathbf{A}\boldsymbol{\theta}}{\sigma^2} \\ &= \frac{\mathbf{A}^T\mathbf{A}}{\sigma^2}\left[(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{p} - \boldsymbol{\theta}\right] \\ &= \mathbf{I}(\boldsymbol{\theta})\left[\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right] \end{aligned} \tag{2.104}$$

which means that the estimator attains the CRB and is efficient. Because the CRB is the lower bound for any unbiased estimator, the estimator is a minimum variance unbiased (MVU) estimator [41]. In conclusion, the estimator derived in this subsection is not only the least square estimator and the MLE but also a MVU estimator which achieves the lower limit - CRB.

2.8.2 Estimation of $p_0(d_0)$, n_p, and σ

If unbiased range estimators are desired, e.g., the proposed LC estimator, σ needs to be known. First let $s = \sigma^2$. Take the first derivative of the log-likelihood function

with respect to s:

$$\frac{\partial \mathcal{L}}{\partial s} = -\frac{q}{2s} + \frac{\sum_{i=1}^{q} \left(p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10 n_p \log_{10} \left(\frac{d_i}{d_0} \right) \right)^2}{2s^2} \tag{2.105}$$

Then the MLE for s is

$$\hat{s} = \hat{\sigma^2} = \frac{\sum_{i=1}^{q} \left(p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10 n_p \log_{10} \left(\frac{d_i}{d_0} \right) \right)^2}{q} \tag{2.106}$$

However, this is only possible when $p_{0,\mathrm{dB}}$ and n_p are given. The realistic case is accomplished by using estimates $\hat{p}_{0,\mathrm{dB}}$ and $\hat{n}_p$, that is,

$$\hat{s} = \hat{\sigma^2} = \frac{\sum_{i=1}^{q} \left(p_{i,\mathrm{dB}} - \hat{p}_{0,\mathrm{dB}} + 10 \hat{n}_p \log_{10} \left(\frac{d_i}{d_0} \right) \right)^2}{q}. \tag{2.107}$$

More partial derivatives related to s are given to investigate the CRB:

$$\frac{\partial^2 \mathcal{L}}{\partial s^2} = \frac{q}{2s^2} - \frac{\sum_{i=1}^{q} \left(p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10 n_p \log_{10} \left(\frac{d_i}{d_0} \right) \right)^2}{s^3} \tag{2.108}$$

$$\frac{\partial^2 \mathcal{L}}{\partial p_{0,\mathrm{dB}} \partial s} = -\frac{\sum_{i=1}^{q} p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10 n_p \log_{10} \left(\frac{d_i}{d_0} \right)}{s^2} = \frac{\partial^2 \mathcal{L}}{\partial s \partial p_{0,\mathrm{dB}}} \tag{2.109}$$

$$\frac{\partial^2 \mathcal{L}}{\partial n_p \partial s} = \frac{\partial^2 \mathcal{L}}{\partial s \partial n_p} = \frac{\sum_{i=1}^{q} \left(p_{i,\mathrm{dB}} - p_{0,\mathrm{dB}} + 10 n_p \log_{10} \left(\frac{d_i}{d_0} \right) \right) 10 \log_{10} \left(\frac{d_i}{d_0} \right)}{s^2} \tag{2.110}$$

The Fisher information matrix for $\boldsymbol{\theta} = [p_{0,\mathrm{dB}}\ n_p\ s]^T$ is

$$\begin{aligned}
\mathbf{I}_3(\boldsymbol{\theta}) &= \begin{bmatrix} -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial p_{0,\mathrm{dB}}^2}\right] & -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial p_{0,\mathrm{dB}} \partial n_p}\right] & -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial p_{0,\mathrm{dB}} \partial s}\right] \\ -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial n_p \partial p_{0,\mathrm{dB}}}\right] & -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial n_p^2}\right] & -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial n_p \partial s}\right] \\ -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial s \partial p_{0,\mathrm{dB}}}\right] & -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial s \partial n_p}\right] & -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial s^2}\right] \end{bmatrix} \\
&= \frac{1}{\sigma^2} \begin{bmatrix} q & -\sum_{i=1}^{q} 10 \log_{10}\left(\frac{d_i}{d_0}\right) & 0 \\ -\sum_{i=1}^{q} 10 \log_{10}\left(\frac{d_i}{d_0}\right) & 100 \sum_{i=1}^{q} \log_{10}^2\left(\frac{d_i}{d_0}\right) & 0 \\ 0 & 0 & \frac{2\sigma^2}{q} \end{bmatrix} \qquad (2.111)
\end{aligned}$$

Then the CRB for $\hat{p}_{0,\mathrm{dB}}$, $\hat{n}_p$, $\hat{s} = \hat{\sigma^2}$ are $[\mathbf{I}_3^{-1}(\boldsymbol{\theta})]_{1,1}$, $[\mathbf{I}_3^{-1}(\boldsymbol{\theta})]_{2,2}$, and $[\mathbf{I}_3^{-1}(\boldsymbol{\theta})]_{3,3}$ respectively. $\mathbf{I}_3(\boldsymbol{\theta})$ is a compound of two block diagonal matrices and the inversion of a block diagonal matrix is the inversion of each small block diagonal matrix. Therefore $[\mathbf{I}_3^{-1}(\boldsymbol{\theta})]_{1,1} = [\mathbf{I}_2^{-1}(\boldsymbol{\theta})]_{1,1}$ and $[\mathbf{I}_3^{-1}(\boldsymbol{\theta})]_{2,2} = [\mathbf{I}_2^{-1}(\boldsymbol{\theta})]_{2,2}$ which means the CRB of $\hat{p}_{0,\mathrm{dB}}$ and $\hat{n}_p$ remain unchanged even though adding another unknown parameter $s = \sigma^2$. Additionally, the CRB for $\hat{s} = \hat{\sigma^2}$ is $2\sigma^4/q$ and it is only achievable if $p_{0,\mathrm{dB}}$ and n_p are known as in (2.106). The above MLE is also an MVU estimator because it attains the CRB.

2.8.3 Location Estimation using Estimated Channel Parameters

After estimating the channel parameters: $\hat{p}_{0,\mathrm{dB}}$, $\hat{n}_p$, and $\hat{\sigma}^2$, it would be desired to use the estimated channel to do localization and then investigate its effect on the accuracy of localization. The simulations consider the same unit circle topology as in Section 2.5; the single blindfolded node is in the origin, and anchors are randomly placed inside the circle. Moreover, the anchors are assumed to sense all nodes inside the circle. The channel parameters thus can be estimated by utilizing the perfectly known distances and noisy power measurements between anchors. Then

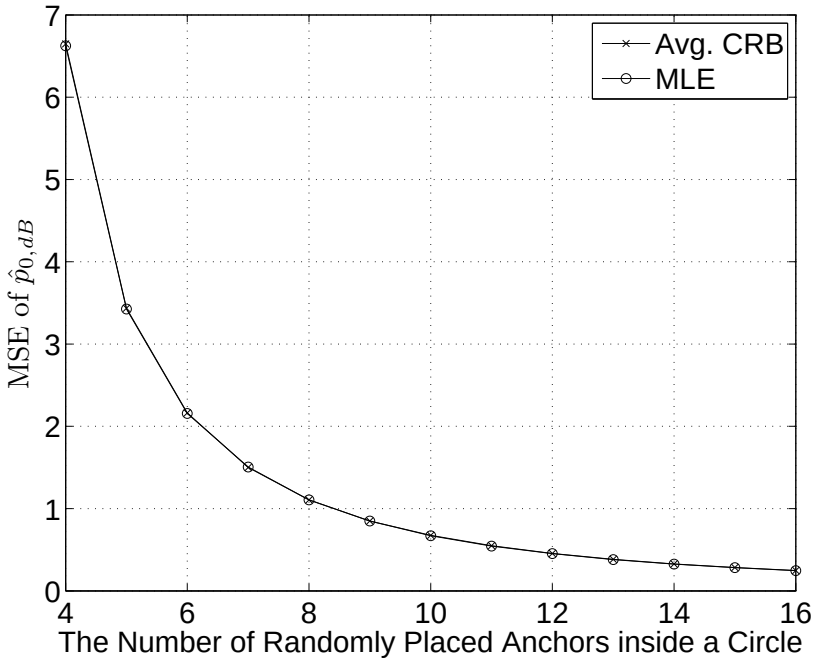

Figure 2.20: MSE of $\hat{p}_{0,\mathrm{dB}}$ versus the number of randomly placed anchors inside a unit circle

the estimated parameters are used for localization instead of the true parameters.

Each topology has a unique CRB and the average CRB over different topologies will be investigated. Again, each topology only has one set of the noisy power measurements. The sample RMSEs for parameters and locations are taken over noisy measurements with different topologies. The simulation results for $\hat{p}_{0,\mathrm{dB}}$, $\hat{n}_p$, and $\hat{\sigma^2}$ are in Figures 2.20 - 2.22. As proven in previous subsections, the CRB for both $\hat{p}_{0,\mathrm{dB}}$ and $\hat{n}_p$ only depends on the number of anchors and their relative distances. Therefore, different channel parameters ($p_{0,\mathrm{dB}}$, n_p, and σ^2) still have the same CRB. The MSEs of joint estimation of $\hat{p}_{0,\mathrm{dB}}$ and $\hat{n}_p$ are also independent of channel parameters, because they are exactly the same as the CRB. The CRB for $\hat{\sigma}^2$ depends on σ and the number of anchors. In all simulations, σ is fixed to be 5 and

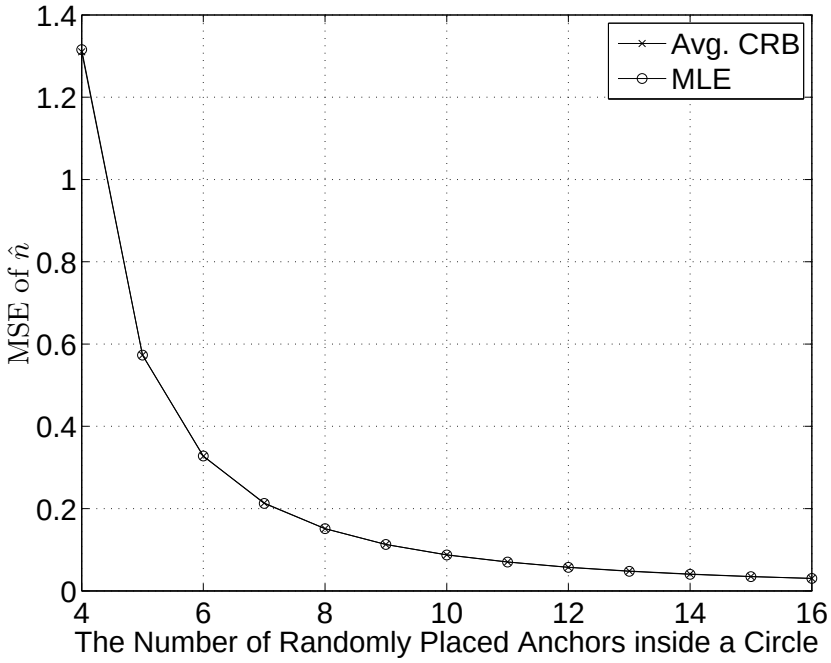

Figure 2.21: MSE of $\hat{n}_p$ versus the number of randomly placed anchors inside a unit circle

n_p is changed to make the SNR different. This makes all related CRBs for $\hat{\sigma}^2$ equal because of the fixed σ. Although the MLE for $\hat{\sigma}^2$ does not achieve the CRB, the MLE is very close to the CRB and almost the same for different n_p. As mentioned earlier, the error of RSS localization is the same as long as the SNR n_p/σ remains unchanged. However, the above argument fails if channel parameters need to be estimated.

In the LC estimator, the scalar factor of unbiased estimate c is defined as

$$c \equiv \exp\left(-\frac{\sigma^2 \ln^2 10}{200 n_p^2}\right) \tag{2.112}$$

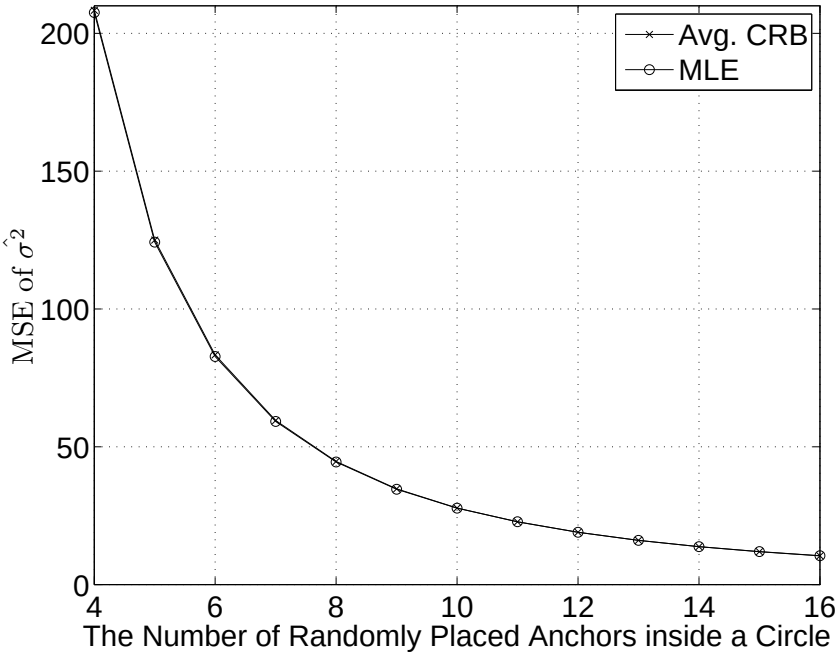

Figure 2.22: MSE of $\hat{\sigma}^2$ versus the number of randomly placed anchors inside a unit circle

and the estimation of c is given by the MLE $\hat{n}_p$ and $\hat{\sigma^2}$

$$\hat{c} \equiv \exp\left(-\frac{\hat{\sigma^2}\ln^2 10}{200\hat{n_p}^2}\right). \tag{2.113}$$

The resulting MSEs for various n_p are presented in Figure 2.23 and larger values of n_p decrease the error. When there are 16 anchors, the MSEs are 0.038, 1.4×10^{-4}, and 1.4×10^{-5} for $n_p = 1.5$, 3, and 5, respectively. Finally, the 10^5 simulation results of the LC estimator and LC estimator using ML ranges with estimated channel parameters are shown in Figures 2.24 - 2.26 for $n_p = 1.5$, 3, and 5, respectively. The LC estimator requires to know parameters: $\hat{p}_{0,\mathrm{dB}}$, $\hat{n}_p$, and $\hat{\sigma^2}$ and the ML ranges only needs the first two parameters. The LC estimator and LC estimator using ML ranges with perfectly known parameters are also given as a comparison. As seen previously, the

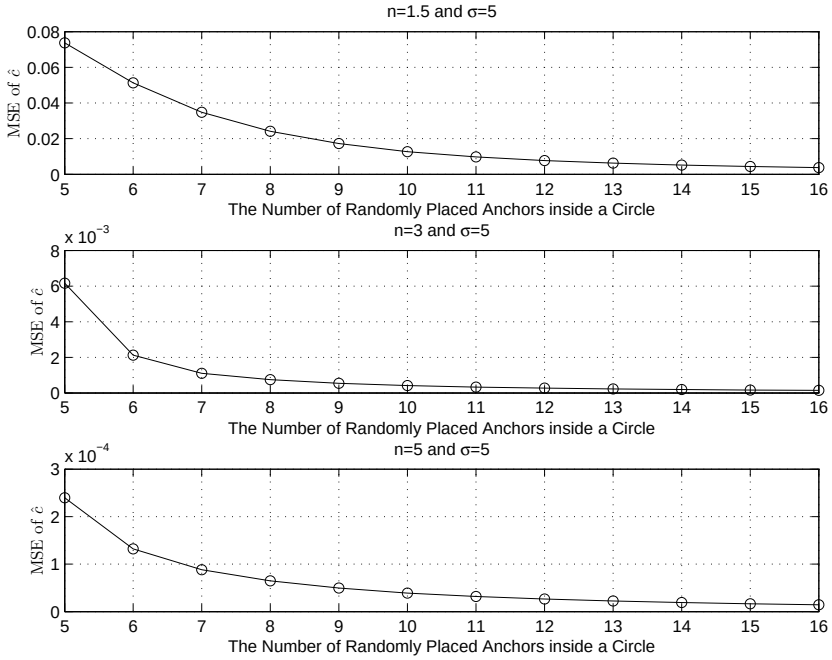

Figure 2.23: MSE of $\hat{c}$ versus the number of randomly placed anchors inside a unit circle

MSE of parameters is large without enough anchors. More importantly, the estimates can be catastrophic which make the range estimates as strong as 0/0. Therefore, a small modification sets $\hat{n}_p = 1.1$ for any $\hat{n}_p < 1.1$. Then $\hat{\sigma^2}$ and $\hat{c}$ are also modified according to the $\hat{n}_p = 1.1$. The MSE of $\hat{n}_p$ still uses the $\hat{n}_p$ before chopped. If the final location estimate remains odd after setting $\hat{n}_p = 1.1$, a random location drawn from $[-1, 1] \times [-1, 1]$ is the final estimate. Under this modification, only 7 times for $n_p = 3$ and 12 times for $n_p = 5$ require the use of a random guess and all of them happen in the case with 3 anchors. For $n_p = 1.5$, there is no random guess in 10^5 trails. The results show that the difference between using perfect parameter and estimated parameters is small if more than 10 anchors are inside the circle. 10 anchors provide 45 pairs of power measurements and relative distances. Moreover,

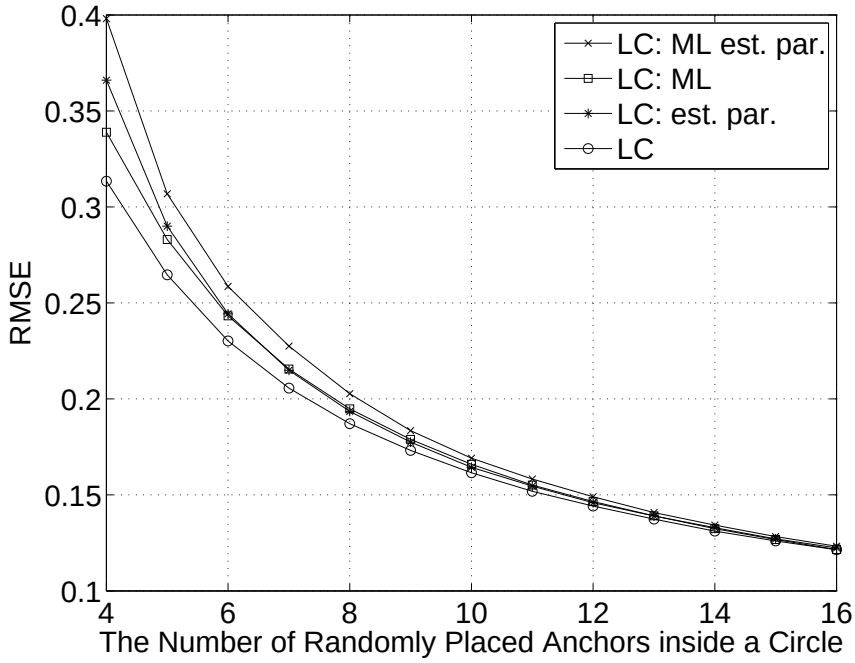

Figure 2.24: RMSE of two LCs with estimated or perfect channel parameters versus the number of randomly placed anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

the larger n_p and smaller σ require fewer anchors to minimize the difference.

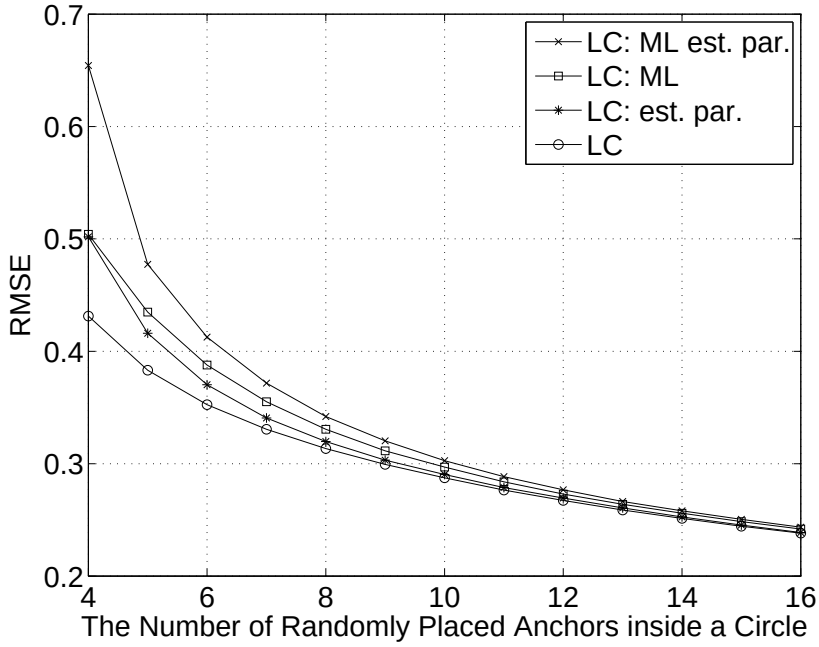

Figure 2.25: RMSE of two LCs with estimated or perfect channel parameters versus the number of randomly placed anchors inside a unit circle with $n_p = 1.5$ and $\sigma = 5$.

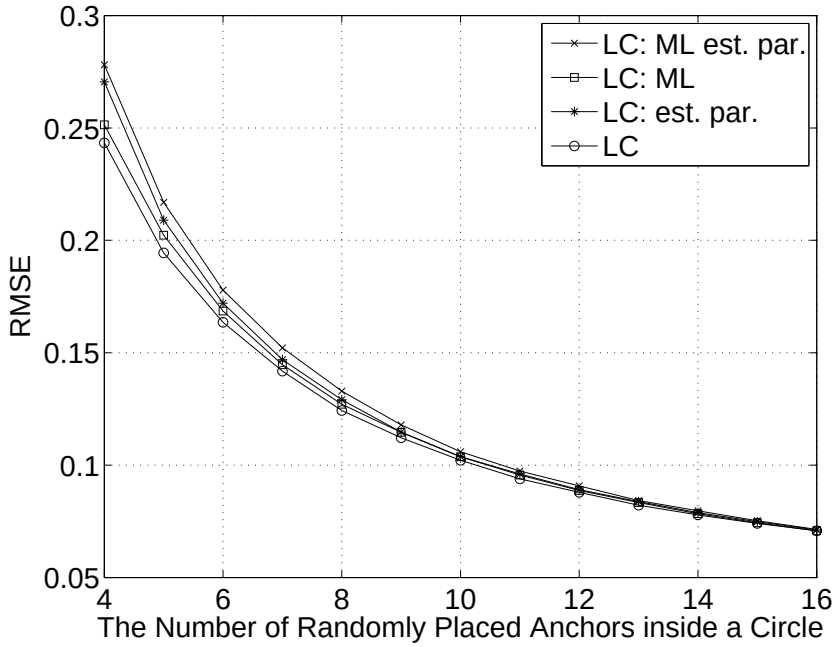

Figure 2.26: RMSE of two LCs with estimated or perfect channel parameters versus the number of randomly placed anchors inside a unit circle with $n_p = 5$ and $\sigma = 5$.

3. COOPERATIVE LOCALIZATION

This section discusses cooperative localization where multiple blindfolded nodes in the observation region can help one another acquire better location estimates [14, 16]. It is assumed that there are m blindfolded nodes and n anchors. Those unknown-location devices are enumerated from 1 to m and the anchors are given the number $m+1, \ldots, m+n$. The blindfolded nodes now are not only receivers but also transmitters which provide RF signals. Moreover, the blindfolded nodes are assumed to have no prior information.

3.1 System Model and Maximum Likelihood Estimator

The received signal strength (RSS) model is still applied for the wireless sensor networks. The following discussions generally apply to all dimensions as well as two dimensions for the purpose of illustration.

Let P_{ij} be the random power transmitted from node i and received by node j. Furthermore, P_{ij} is governed by the same log-normal shadowing as was used for range estimators and non-cooperative localization. That is,

$$P_{ij} = p_0 \left(\frac{d_{ij}}{d_0} \right)^{-n_p} 10^{\frac{Z_{ij}}{10}} \tag{3.1}$$

for $i = 1, \ldots, m+n$ and $j = 1, \ldots, m$. All above parameters are defined as before and the Z_{ij}'s are independent and identically distributed (i.i.d.) zero mean Gaussian random variables with variance σ^2. The Z_{ij} is assumed to be equl to Z_{ji} because the signal transmitted from i to j has the same environment with the signal transmitted from j to i. That results in $P_{ij} = P_{ji}$, the reciprocal property.

Let $\rho_{ij} = p_{0,\mathrm{dB}} - 10 n_p \log_{10}(\frac{d_{ij}}{d_0})$ where $d_{ij} = \|\theta_j - \theta_i\|$ is the distance between

node i and node j, and $d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$ in 2D. The conditional PDF in this system is written as

$$f_{\mathbf{P}_{\mathrm{dB}}}(\mathbf{p}_{\mathrm{dB}}|\boldsymbol{\Theta} = \boldsymbol{\theta}) = \prod_{i=2}^{m+n} \prod_{j=1}^{\min\{i-1,m\}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(p_{ij,\mathrm{dB}} - \rho_{ij})^2}{2\sigma^2}\right) \tag{3.2}$$

where $\mathbf{p}_{\mathrm{dB}}$ is the vector containing $nm + m(m-1)/2$ pairwise received power measurements in decibels, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_{m+n})$ is the location vector for m unknown-location nodes and n anchors, and $\theta_i = [x_i, y_i]$ is the location coordinate of node i in two dimensions.

A wireless network could be very large and the transmitted powers are usually limited. One device may not hear all other devices in the network . Thus h_{ij} is introduced to represent the existence of direct wireless link between i and j, i.e.,

$$h_{ij} = \begin{cases} 1 & \text{if } P_{ij} > p_{\mathrm{th}} \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

for $i = m+1, \ldots, m+n$, and $j = 1, \ldots, m$. p_{th} is the minimum received power required for reception. Moreover, h_{ii} is defined by 0 because node i does not connect to itself.

3.2 Maximum Likelihood Estimator by Coordinate Descent Method

Similarly to the previous two sections, the maximum likelihood estimator (MLE) and its family are discussed in the cooperative case. The MLE can be expressed as

$$\begin{aligned} \hat{\boldsymbol{\theta}}_{\mathrm{ML}}(\mathbf{p}) &= \underset{\boldsymbol{\vartheta} \in \mathcal{S}}{\operatorname{argmin}} \sum_{i=2}^{m+n} \sum_{j=1}^{\min\{i-1,m\}} h_{ij} \left[\log\left(\frac{p_{ij}}{p_0}\left(\frac{\|\vartheta_j - \vartheta_i\|}{d_0}\right)^{n_p}\right)\right]^2 \\ &= \underset{\boldsymbol{\vartheta} \in \mathcal{S}}{\operatorname{argmin}} \sum_{i=2}^{m+n} \sum_{j=1}^{\min\{i-1,m\}} h_{ij} \left[\log\left(\frac{d(\vartheta_i, \vartheta_j)}{\hat{d}_{ij,\mathrm{ML}}}\right)\right]^2 \end{aligned} \tag{3.4}$$

where $\mathbf{p} = [p_{21}, \ldots, p_{m+n,m}]$ is a vector containing $nm + m(m-1)/2$ received power measurements, $\boldsymbol{\vartheta} = [\vartheta_1, \ldots, \vartheta_{n+m}]$ is a candidate vector, and $\mathcal{S}$ represents a search space. $\vartheta_i = \theta_i$ for $i = m+1, \ldots, m+n$ because the locations of anchors are known. Moreover, $\hat{d}_{ij,\mathrm{ML}} = d_0(p_{ij}/p_0)^{-1/n_p}$ is the ML range estimate and $d(\vartheta_i, \vartheta_j) = \|\vartheta_j - \vartheta_i\|$ represents the distance between candidates ϑ_i and ϑ_j. In two dimensions, the MLE can be written as:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{\mathrm{ML}}(\mathbf{p}) &= \underset{\boldsymbol{\vartheta}\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=2}^{m+n} \sum_{j=1}^{\min\{i-1,m\}} h_{ij} \left[\log\left(\frac{p_{ij}}{p_0} \left(\frac{\sqrt{(\xi_j - \xi_i)^2 + (\psi_j - \psi_i)^2}}{d_0} \right)^{n_p} \right) \right]^2 \\
&= \underset{\boldsymbol{\vartheta}\in\mathcal{S}}{\operatorname{argmin}} \sum_{i=2}^{m+n} \sum_{j=1}^{\min\{i-1,m\}} h_{ij} \left[\log\left(\frac{d([\xi_i, \psi_i], [\xi_j, \psi_j])}{\hat{d}_{ij,\mathrm{ML}}} \right) \right]^2 \qquad (3.5)
\end{aligned}$$

where $d([\xi_i, \psi_i], [\xi_j, \psi_j]) = \sqrt{(\xi_j - \xi_i)^2 + (\psi_j - \psi_i)^2}$ represents the distance between candidates $[\xi_i, \psi_i]$ and $[\xi_j, \psi_j]$. The candidate vector becomes $\boldsymbol{\vartheta} = [\xi_1, \psi_1, \ldots, \xi_{n+m}, \psi_{n+m}]$ in 2D. Besides the received powers and anchors' locations, the MLE requires the channel parameters n_p and $p_0(d_0)$. The above cost function is still non-convex as with the non-cooperative case.

The exhaustive search used in the single-unknown case becomes infeasible in a multiple-unknown situation since the computation increases exponentially as m increases. Therefore, some optimization algorithms, e.g., gradient methods, are required to solve these problems [53]. The main idea of gradient methods is to iteratively approach the local minima based on the gradient of the current point.

The *coordinate descent (CD) method* is another possible approach to determine the minimum of a cost function [54, 53]. The idea is to minimize a cost function with respect to one coordinate at a time until certain constraints are satisfied. Clearly, the CD method is a distributed algorithm. A two-dimensional case is discussed here and other dimension can be extended similarly. $[\hat{x}_i^{(k+1)}, \hat{y}_i^{(k+1)}]$ represents the estimated

coordinate of node i $(i = 1, \ldots, m+n)$ at the $(k+1)$-th stage and it is determined by

$$\hat{x}_i^{(k+1)} = \underset{\xi \in \mathcal{S}_{x_i}}{\operatorname{argmin}} f_i(\hat{x}_1^{(k+1)}, \hat{y}_1^{(k+1)}, \ldots, \hat{x}_{i-1}^{(k+1)}, \hat{y}_{i-1}^{(k+1)}, \xi, \hat{y}_i^{(k)}, \hat{x}_{i+1}^{(k)}, \ldots, \hat{x}_{m+n}^{(k)}, \hat{y}_{m+n}^{(k)}), \tag{3.6}$$

$$\hat{y}_i^{(k+1)} = \underset{\psi \in \mathcal{S}_{y_i}}{\operatorname{argmin}} g_i(\hat{x}_1^{(k+1)}, \hat{y}_1^{(k+1)}, \ldots, \hat{y}_{i-1}^{(k+1)}, \hat{x}_i^{(k+1)}, \psi, \hat{x}_{i+1}^{(k)}, \hat{y}_{i+1}^{(k)}, \ldots, \hat{x}_{m+n}^{(k)}, \hat{y}_{m+n}^{(k)}) \tag{3.7}$$

Two cost functions f_i and g_i are defined as follows by ignoring iteration notation k or $k+1$ for simplicity:

$$f_i(\cdots) = \underset{\xi \in \mathcal{S}_{x_i}}{\operatorname{argmin}} \sum_{j=1}^{m+n} h_{ij} \left[\log \left(\frac{p_{ij}}{p_0} \left(\frac{\sqrt{(\hat{x}_j - \xi)^2 + (\hat{y}_j - \hat{y}_i)^2}}{d_0} \right)^{n_p} \right) \right]^2, \tag{3.8}$$

$$g_i(\cdots) = \underset{\psi \in \mathcal{S}_{y_i}}{\operatorname{argmin}} \sum_{j=1}^{m+n} h_{ij} \left[\log \left(\frac{p_{ij}}{p_0} \left(\frac{\sqrt{(\hat{x}_j - \hat{x}_i)^2 + (\hat{y}_j - \psi)^2}}{d_0} \right)^{n_p} \right) \right]^2 \tag{3.9}$$

where $\mathcal{S}_{x_i}$ and $\mathcal{S}_{y_i}$ are the search spaces for $\hat{x}_i$ and $\hat{y}_i$, respectively. For the anchors $(i = m+1, \ldots, m+n)$, $\hat{x}_i^{(k)} = x_i$ and $\hat{y}_i^{(k)} = y_i$ for all k. There are many ways to assign the initial location estimates $\hat{x}_i^{(0)}$ and $\hat{y}_i^{(0)}$, e.g., the discrete exhaustive search in non-cooperative case. This dissertation applies CD to initialize but only uses anchors' locations and powers if at least one anchor connects to the blindfolded i. If node i connects to no anchors, then random numbers are assigned for the initial estimates. The initial procedure is also a non-cooperative localization. This distributed optimization can be executed by an exhaustive search while optimizing each individual location. Again, the exhaustive search is more accurate than gradient methods in this non-convex optimization problem for RSS localization.

The coordinate descent method is a special case of nonlinear Gsuss-Seidel algo-

rithm which successively minimizes each component [55]. In contrast, each component is simultaneously minimized in the nonlinear Jacobi algorithm.

3.3 Linear Combination Estimator

This section derives the cooperative linear combination (LC) estimator using the concept of the best linear unbiased estimator (BLUE) [41] where each blindfolded node combines multiple ML-based location estimates from other unknown-location nodes as well as the anchors. Similar to the non-cooperative LC, a location estimate is computed with a transmitter's location, a range estimate, and a direction. The receiver then combines all its obtained location estimates to estimate its location. The only difference is that a transmitter's location needs to be estimated if the transmitter is another blindfolded node. Since the estimated location of one blindfolded node can be improved after receiving other nodes' information, all blindfolded nodes continue exchanging information until the performance stabilizes. Therefore, an iterative refinement will be applied to the cooperative linear combination estimator.

3.3.1 Linear Combination Estimator under Perfect Directions

To derive the BLUE algorithm, the directions among nodes are first assumed to be perfectly known by corresponding receivers. Under this assumption, the optimal combining weights are derived. In addition, the resulting MSE based on optimal weights and perfect directions, which is also the variance, will be a lower bound of MSE with any angle/direction estimations. Since the optimal MSE/variance will be useful for conducting analysis without many simulations for a fixed tologoy, the subsection is also named the analytical part for linear combination estimator.

As mentioned before, iterations are used to refine the location estimates. After

k iterations, the estimate of node j is

$$\hat{\theta}_j^{(k+1)} = \sum_{i=1}^{m+n} a_{ij}^{(k)} \hat{\theta}_{ij}^{(k)} \quad \text{for} \quad j = 1, \ldots, m \tag{3.10}$$

where $\hat{\theta}_{ij}^{(k)}$ is the location estimate of node j based on node i and $a_{ij}^{(k)}$ is the combining weight in the k-th step. Furthermore, $0 \leq a_{ij} \leq 1$ and $\sum a_{ij} = 1$ will be desired as in the case of non-cooperative localization. It is natural to let $a_{jj}^{(k)} = 0$ since a node contributes nothing to itself unless it has prior knowledge. This will be more clear when the optimal weights are derived subsequently. The $\hat{\theta}_{ij}^{(k)}$ are computed by using the estimated location of node i without node j's information $\hat{\omega}_{ij}^{(k)}$, the range estimates $\hat{d}_{ij}$, and the the direction $\mathbf{v}_{ij}$ from node i to node j. In other words,

$$\hat{\theta}_{ij}^{(k)} = \hat{\omega}_{ij}^{(k)} + \hat{d}_{ij} \mathbf{v}_{ij} \tag{3.11}$$

for $i = 1, \ldots, m+n$, and $j = 1, \ldots, m$. The true direction is given by

$$\mathbf{v}_{ij} = \frac{\theta_j - \theta_i}{\|\theta_j - \theta_i\|} = \frac{\theta_j - \theta_i}{d_{ij}}, \tag{3.12}$$

and $0 \times \infty = 0$ (or $0/0 = 0$) as a convention. In 2D, the true direction can be expressed as

$$\mathbf{v}_{ij} = \frac{[x_j - x_i, y_j - y_i]}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \tag{3.13}$$

The $\hat{\omega}_{ij}^{(k)}$ is similar to $\hat{\theta}_i^{(k)} = \sum_{l=1}^{m+n} a_{li}^{(k-1)} \hat{\theta}_{li}^{(k-1)}$ and the only difference is to exclude the information from node j since it will provide information to node j. Therefore

for m blindfolded nodes:

$$\begin{aligned}\hat{\omega}_{ij}^{(k)} &= \frac{1}{1-a_{ji}^{(k-1)}} \sum_{l=1,l\neq j}^{m+n} a_{li}^{(k-1)} \hat{\theta}_{li}^{(k-1)} = \frac{1}{1-a_{ji}^{(k-1)}} \sum_{l=1,l\neq j}^{m+n} a_{li}^{(k-1)} (\hat{\omega}_{li}^{(k-1)} + \hat{d}_{li}\mathbf{v}_{li}) \\ &= \frac{\hat{\theta}_i^{(k)} - a_{ji}^{(k-1)} (\hat{\omega}_{ji}^{(k-1)} + \hat{d}_{ji}\mathbf{v}_{ji})}{1-a_{ji}^{(k-1)}} \quad \text{for} \quad i = 1, \ldots, m. \end{aligned} \tag{3.14}$$

For n anchors:

$$\hat{\omega}_{ij}^{(k)} = \theta_i \quad \text{for} \quad i = m+1, \ldots, m+n \tag{3.15}$$

which are the true locations. For the special case $i = j$, $\hat{\omega}_{jj}^{(k)} = \hat{\theta}_j^{(k)}$ because $a_{jj} = 0$. Then $\hat{\theta}_{jj}^{(k)} = \hat{\theta}_j^{(k)}$ since $\hat{d}_{jj}$ and $\mathbf{v}_{jj}$ are both zero.

If an unbiased location estimate of node i and an unbiased range estimator are applied, i.e., $\mathbb{E}[\hat{\omega}_{ij}^{(k)}] = \theta_i$ and $\mathbb{E}[\hat{d}_{ij}] = d_{ij}$,

$$\mathbb{E}[\hat{\theta}_{ij}^{(k)}] = \mathbb{E}[\hat{\omega}_{ij}^{(k)} + \hat{d}_{ij}\mathbf{v}_{ij}] = \theta_i + d_{ij}\frac{\theta_j - \theta_i}{d_{ij}} = \theta_j. \tag{3.16}$$

In other words, $\hat{\theta}_{ij}^{(k)}$ is also unbiased. Applying the Gauss-Markov theorem, we can have the best linear unbiased estimator (BLUE) [41] for $\hat{\theta}_j^{(k+1)}$ among those $m+n$ unbiased estimates:

$$\hat{\theta}_{j,\text{opt}}^{(k+1)} = \sum_{i=1}^{m+n} a_{ij,\text{opt}}^{(k)} \hat{\theta}_{ij}^{(k)} \quad \text{for} \quad j = 1, \ldots, m. \tag{3.17}$$

The variance of BLUE which is the minimum variance of any possible unbiased $\hat{\theta}_j^{(k+1)}$ and its corresponding optimal weight vector are

$$\text{Var}(\hat{\theta}_{j,\text{opt}}^{(k+1)}) = \frac{1}{\mathbf{s}^T \left(\mathbf{C}_j^{(k)}\right)^{-1} \mathbf{s}} \tag{3.18}$$

and

$$\mathbf{a}_{j,\text{opt}}^{(k)} = \left[a_{1j,\text{opt}}^{(k)}, \cdots, a_{(m+n)j,\text{opt}}^{(k)}\right]^{\text{T}} = \frac{\left(\mathbf{C}_j^{(k)}\right)^{-1}\mathbf{s}}{\mathbf{s}^T\left(\mathbf{C}_j^{(k)}\right)^{-1}\mathbf{s}} \tag{3.19}$$

where $\mathbf{s} = [1 \cdots 1]^T$ is the scaled mean vector and $\mathbf{C}_j^{(k)}$ is the covariance matrix of the location estimates for node j in the k-th step, i.e., $[\mathbf{C}_j^{(k)}]_{il} = \mathbb{E}[(\hat{\theta}_{ij}^{(k)} - \theta_j)(\hat{\theta}_{lj}^{(k)} - \theta_j)]$.

If the corresponding location estimates $\hat{\theta}_{ij}^{(k)}$ are all uncorrelated, the covariance matrix is diagonal: $\mathbf{C}_j^{(k)} = \text{diag}\left(\text{Var}(\hat{\theta}_{1,j}^{(k)}), \ldots, \text{Var}(\hat{\theta}_{m+n,j}^{(k)})\right)$. Assume node j receives the information from both blindfolded node i and l, and they provide their location estimates as $\hat{\theta}_i^{(k)}$ and $\hat{\theta}_l^{(k)}$ instead of $\hat{\omega}_{ij}^{(k)}$ and $\hat{\omega}_{lj}^{(k)}$, respectively. Then there exists dependence between nodes i and l because they both have $\hat{\theta}_j^{(k-1)}$.

The variance of a vector is defined to be a scalar as

$$\text{Var}(\hat{\theta}_{ij}^{(k)}) \equiv \mathbb{E}\left[\left\|\hat{\theta}_{ij}^{(k)} - \mathbb{E}[\hat{\theta}_{ij}^{(k)}]\right\|^2\right] = \mathbb{E}\left[\left\|\hat{\theta}_{ij}^{(k)} - \theta_j\right\|^2\right], \tag{3.20}$$

In 2-D case, let $\hat{\theta}_{ij}^{(k)} = [\hat{x}_{ij}^{(k)}, \hat{y}_{ij}^{(k)}]$ and $\hat{\omega}_{ij}^{(k)} = [\hat{\xi}_{ij}^{(k)}, \hat{\psi}_{ij}^{(k)}]$. The variance becomes

$$\text{Var}(\hat{\theta}_{ij}^{(k)}) = \mathbb{E}\left[(\hat{x}_{ij}^{(k)} - x_j)^2 + (\hat{y}_{ij}^{(k)} - y_j)^2\right] = \mathbb{E}\left[(\hat{x}_{ij}^{(k)})^2\right] - x_j^2 + \mathbb{E}\left[(\hat{y}_{ij}^{(k)})^2\right] - y_j^2 \tag{3.21}$$

where

$$\begin{aligned}
\mathbb{E}\left[(\hat{x}_{ij}^{(k)})^2\right] &= \mathbb{E}\left[\left(\hat{\xi}_{ij}^{(k)} + \hat{d}_{ij}\frac{x_j - x_i}{d_{ij}}\right)^2\right] \\
&= \mathbb{E}\left[(\hat{\xi}_{ij}^{(k)})^2\right] + \mathbb{E}[\hat{d}_{ij}^2]\frac{(x_j - x_i)^2}{d_{ij}^2} + 2\mathbb{E}[\hat{\xi}_{ij}^{(k)}]\mathbb{E}[\hat{d}_{ij}]\frac{x_j - x_i}{d_{ij}} \\
&= \mathbb{E}\left[(\hat{\xi}_{ij}^{(k)})^2\right] + \mathbb{E}[\hat{d}_{ij}^2]\frac{(x_j - x_i)^2}{d_{ij}^2} + 2x_i x_j - 2x_i^2, \qquad (3.22) \\
\mathbb{E}\left[(\hat{y}_{ij}^{(k)})^2\right] &= \mathbb{E}\left[(\hat{\psi}_{ij}^{(k)})^2\right] + \mathbb{E}[\hat{d}_{ij}^2]\frac{(y_j - y_i)^2}{d_{ij}^2} + 2y_i y_j - 2y_i^2. \qquad (3.23)
\end{aligned}$$

$\mathbb{E}[\hat{\xi}_{ij}^{(k)}\hat{d}_{ij}] = \mathbb{E}[\hat{\xi}_{ij}^{(k)}]\mathbb{E}[\hat{d}_{ij}]$ is given by $\hat{\omega}_{ij}^{(k)}$ and $\hat{d}_{ij}$ are uncorrelated. On the other hand, if the information of node i is not extracted, $\hat{\omega}_{ij}^{(k)}$ contains $\hat{d}_{ji}$ which is the same as $\hat{d}_{ij}$ because of the reciprocity. Then they are definitely correlated. It should be noticed that $\hat{\omega}_{ij}^{(k)}$ may have the information $\hat{d}_{ij}$ or $\hat{d}_{ji}$ indirectly from other nodes. However, extracting $\hat{d}_{ji}$ is the simplest way to make $\hat{\omega}_{ij}^{(k)}$ and $\hat{d}_{ij}$ uncorrelated as possible as it can be. Then the variance of node j's estimate utilized node j in 2D is

$$\begin{aligned}
\mathrm{Var}(\hat{\theta}_{ij}^{(k)}) &= \mathbb{E}\left[(\hat{\xi}_{ij}^{(k)})^2\right] - x_i^2 + \mathbb{E}\left[(\hat{\psi}_{ij}^{(k)})^2\right] - y_i^2 \\
&\quad + \mathbb{E}[\hat{d}_{ij}^2] + 2x_ix_j + 2y_iy_j - x_i^2 - y_i^2 - x_j^2 - y_j^2 \\
&= \mathbb{E}\left[(\hat{\xi}_{ij}^{(k)})^2\right] - \mathbb{E}^2[\hat{\xi}_{ij}^{(k)}] + \mathbb{E}\left[(\hat{\psi}_{ij}^{(k)})^2\right] - \mathbb{E}^2[\hat{\psi}_{ij}^{(k)}] + \mathbb{E}[\hat{d}_{ij}^2] - (x_j - x_i)^2 - (y_j - y_i)^2 \\
&= \mathrm{Var}(\hat{\xi}_{ij}^{(k)}) + \mathrm{Var}(\hat{\psi}_{ij}^{(k)}) + \mathbb{E}[\hat{d}_{ij}^2] - d_{ij}^2 \\
&= \mathrm{Var}(\hat{\omega}_{ij}^{(k)}) + \mathrm{Var}(\hat{d}_{ij}) = \mathrm{Var}(\hat{\omega}_{ij}^{(k)}) + [\exp(b^2) - 1]d_{ij}^2/h_{ij}. \qquad (3.24)
\end{aligned}$$

where h_{ij} indicates whether wireless connection exists between nodes i and j (as $h_{ij} = 1$ for connection) in (3.3). $\mathrm{Var}(\hat{\theta}_{ij}^{(k)}) = \mathrm{Var}(\hat{\omega}_{ij}^{(k)}) + \mathrm{Var}(\hat{d}_{ij})$ is still true for other dimensions with unbiased range estimates and known directions. The $\mathrm{Var}(\hat{\theta}_{ij}^{(k)}) = \infty$ if node i is not estimated after k iterations ($\mathrm{Var}(\hat{\omega}_{ij}^{(k)}) = \infty$) or there is no connection between nodes i and j ($h_{ij} = 0$).

Therefore, when a node sends its location estimates by excluding the information from its receiver, it could make both covariance matrix $\mathbf{C}_j^{(k)}$ diagonal and $\mathrm{Var}(\hat{\theta}_{ij}^{(k)}) = \mathrm{Var}(\hat{\omega}_{ij}^{(k)}) + \mathrm{Var}(\hat{d}_{ij})$ in certain levels. In other words, if a node includes information from its receiver, both uncorrelated properties fail.

As mentioned in the previous section, the variances of estimates in both range estimator and non-cooperative LC with true directions can be expressed in the format with a scalar $\exp(b^2) - 1$. It will be convenient if the above expression $\mathrm{Var}(\hat{\theta}_{ij}^{(k)})$ has

the same format. Thus a new terminology called an *effective distance* in the k-th stage iteration is defined as

$$\delta_{ij}^{(k)} = \sqrt{\frac{\mathrm{Var}(\hat{\omega}_{ij}^{(k)})}{\exp(b^2)-1} + \frac{d_{ij}^2}{h_{ij}}} \quad \text{for} \quad k = 0, 1, \ldots \tag{3.25}$$

Then the optimal variance can be expressed as

$$\mathrm{Var}(\hat{\theta}_{ij}^{(k)}) \equiv [\exp(b^2)-1](\delta_{ij}^{(k)})^2 \tag{3.26}$$

which has the similar structure in the non-cooperative case: $\mathrm{Var}(\hat{\theta}_{i0}) = \mathrm{Var}(\hat{d}_i) = [\exp(b^2)-1]d_i^2$. In other words, effective distances in cooperative LC will have the same role as the conventional distances in a non-cooperative one.

The variance of node j under the uncorrelated assumption of coming estimates in the previous stage is given by

$$\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k+1)}) = \frac{\exp(b^2)-1}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(k)})^2} \tag{3.27}$$

and the optimal weights are

$$a_{ij,\mathrm{opt}}^{(k)} = \frac{1/(\delta_{ij}^{(k)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(k)})^2}. \tag{3.28}$$

Obviously, the optimal MSE and weights only depend on distances and are irrelevant to angles. Similarly,

$$\mathrm{Var}(\hat{\omega}_{ji}^{(k+1)}) = \frac{\exp(b^2)-1}{\sum_{l=1,l\neq j}^{m+n} 1/(\delta_{lj}^{(k)})^2}. \tag{3.29}$$

Then the effective distance can be written as

$$\delta_{ij}^{(k)} = \sqrt{\frac{1}{\sum_{l=1,l\neq j}^{m+n} \frac{1}{(\delta_{li}^{(k-1)})^2}} + \frac{d_{ij}^2}{h_{ij}}} \quad \text{for} \quad k = 1, 2, \ldots, \tag{3.30}$$

and

$$\delta_{ij}^{(0)} = \begin{cases} \infty & \text{for} \quad i = 1, \ldots, m \\ \frac{d_{ij}}{h_{ij}} & \text{for} \quad i = m+1, \ldots, m+n \end{cases} . \tag{3.31}$$

There are several properties and observations about effective distances. First of all, the effective distance between two nodes is ∞ if no connection exists between them. Most importantly, the effective distance relates not only to the conventional distance d_{ij} between node i and node j, but also to an additional term $1/\sum_{l\neq j} \frac{1}{(\delta_{li}^{(k-1)})^2}$ which penalizes the unreliable location estimate of blindfold node i. If the blindfolded node i has prior knowledge, the penalty can be assigned to a smaller value related to its uncertainty. On the other hand, the effective distance $\delta_{ij}^{(k)}$ is equal to the Euclidean distance d_{ij} if node i is an anchor for any iteration step k. Consequently, if blindfolded node i and an anchor are at the same position, node i is virtually or "effectively" farther than the anchor. This is the reason to use the term effective distance in (3.25). Therefore, the estimate based on the "effectively" closer node is assigned a higher weight because it provides the smaller estimate error to the receiver. As expected, node i contributes nothing to estimate node j if $\delta_{ij}^{(k)} = \infty$.

To simplify the notation, two new variables are defined as

$$\gamma_j^{(k)} \equiv \frac{1}{\sum_{l=1}^{m+n} \frac{1}{(\delta_{lj}^{(k)})^2}} \tag{3.32}$$

and

$$\gamma_{ji}^{(k)} \equiv \frac{\exp(b^2)-1}{\sum_{l=1,l\neq j}^{m+n} 1/(\delta_{lj}^{(k)})^2}. \tag{3.33}$$

which are actually the penalties for location estimate of node j. Then

$$\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k+1)}) = [\exp(b^2)-1]\gamma_j^{(k)}, \tag{3.34}$$

$$\mathrm{Var}(\hat{\omega}_{ji}^{(k+1)}) = [\exp(b^2)-1]\gamma_{ji}^{(k)}, \tag{3.35}$$

$$\delta_{ij}^{(k)} = \sqrt{\gamma_{ij}^{(k-1)} + \frac{d_{ij}^2}{h_{ij}}}, \tag{3.36}$$

$$a_{ij,\mathrm{opt}}^{(k)} = \frac{\gamma_j^{(k)}}{(\delta_{ij}^{(k)})^2}. \tag{3.37}$$

Moreover, let

$$\eta_j^{(k)} \equiv \frac{1}{\gamma_j^{(k)}} = \sum_{l=1}^{m+n} \frac{1}{(\delta_{lj}^{(k)})^2}, \tag{3.38}$$

$$\eta_{ji}^{(k)} \equiv \frac{1}{\gamma_{ji}^{(k)}} = \sum_{l=1,l\neq i}^{m+n} \frac{1}{(\delta_{lj}^{(k)})^2} \tag{3.39}$$

Then

$$\eta_{ji}^{(k)} = \eta_j^{(k)} - \frac{1}{(\delta_{ij}^{(k)})^2} \tag{3.40}$$

which reduces the computations.

The following paragraphs explicitly describe how the proposed algorithm works step by step. In the initial (0-th) stage, only the anchors know their locations which are actually the true locations, i.e., $\hat{\omega}_{ij}^{(0)} = \hat{\theta}_i^{(0)} = \theta_i$ for $i = m+1, \ldots, m+n$. Node j's estimates based on node i becomes

$$\hat{\theta}_{ij}^{(0)} = \theta_i + \hat{d}_{ij}\mathbf{v}_{ij} \tag{3.41}$$

and

$$\mathrm{Var}(\hat{\theta}_{ij}^{(0)}) = 0 + \mathrm{Var}(\hat{d}_{ij}) = [\exp(b^2) - 1]d_{ij}^2/h_{ij} \tag{3.42}$$

for $i = m+1, \ldots, m+n$ and $j = 1, \ldots, m$. In addition, the effective distance in the initial stage is $\delta_{ij}^{(0)} = d_{ij}/h_{ij}$ for $i = m+1, \ldots, m+n$ because anchors have no estimation error and thus have no penalty. On the other hand, let $\mathrm{Var}(\hat{\omega}_{ij}^{(0)}) = \infty$ which results $\delta_{ij}^{(0)} = \infty$ for $i = 1, \ldots, m$ since all blindfolded nodes are not estimated. Therefore the variance of the optimal $\hat{\theta}_j^{(1)}$ (having the optimal weights) is

$$\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(1)}) = \frac{1}{\mathbf{s}^T \left(\mathbf{C}_j^{(0)}\right)^{-1} \mathbf{s}} = \frac{\exp(b^2) - 1}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(0)})^2} = \frac{\exp(b^2) - 1}{\sum_{l=m+1}^{m+n} h_{lj}/d_{lj}^2} \tag{3.43}$$

and the optimal weights are

$$a_{ij,\mathrm{opt}}^{(0)} = \frac{1/(\delta_{ij}^{(0)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(0)})^2} = \frac{h_{ij}/d_{ij}^2}{\sum_{l=m+1}^{m+n} h_{lj}/d_{lj}^2} \tag{3.44}$$

for $i = m+1, \ldots, m+n$, and $j = 1, \ldots, m$. The above result is the same as the non-cooperative linear combination estimator as in Section 2.3 for the node directly connecting to at least one anchor. The non-cooperative LC can be viewed as a special case of the cooperative LC estimator. Because blindfolded nodes do not communicate in this stage, the information for the next iteration provided by the blindfolded node j is generated as

$$\eta_{ji}^{(0)} = \eta_j^{(0)} = \sum_{l=m+1}^{m+n} \frac{h_{lj}}{d_{lj}^2} \tag{3.45}$$

and

$$\gamma_{ji}^{(0)} = \gamma_j^{(0)} = \frac{1}{\eta_j^{(0)}} = \frac{1}{\sum_{l=m+1}^{m+n} \frac{h_{lj}}{d_{lj}^2}}. \tag{3.46}$$

If a blindfold node connects to no anchors, its variance remains ∞, i.e., $\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(1)}) =$

∞, after the initial stage. That node then will result in contributing nothing to other nodes in the next stage which makes its effective distances to all others as ∞ (penalty $\gamma_{ji}^{(0)} = \infty$, too) and all its corresponding weights are assigned to be 0 by convention.

When the iteration enters the 1-st stage, the blindfolded nodes which obtain their location estimates in the initial stage can help other blindfolded nodes to locate themselves. An estimate provider (node i) only has information from anchors and the anchors have no need to be a receiver. Therefore no information needs to be extracted when a provider sends its estimated location, i.e., $\hat{\omega}_{ij}^{(1)} = \hat{\theta}_i^{(1)}$ for $i = 1, \ldots, m+n$ and $j = 1, \ldots, m$. The variance of node j's estimate provided by node i

$$\mathrm{Var}(\hat{\theta}_{ij}^{(1)}) = \mathrm{Var}(\hat{\omega}_{ij}^{(1)}) + \mathrm{Var}(\hat{d}_{ij}) = \mathrm{Var}(\hat{\theta}_i^{(1)}) + \mathrm{Var}(\hat{d}_{ij}) = [\exp(b^2) - 1](\delta_{ij}^{(1)})^2 \quad (3.47)$$

where the effective distance in the 1-st stage is

$$\delta_{ij}^{(1)} = \sqrt{\gamma_{ij}^{(0)} + \frac{d_{ij}^2}{h_{ij}}} = \sqrt{\frac{1}{\sum_{l=m+1}^{m+n} \frac{h_{li}}{d_{li}^2}} + \frac{d_{ij}^2}{h_{ij}}}. \quad (3.48)$$

The penalty term depends on the relative distances between the estimate provider (node j) and its connected anchors. No other blindfolded nodes' information appears in the penalty because a blindfolded node can estimate itself by only using anchor's information in this stage. The variance of the optimal $\hat{\theta}_j^{(2)}$ is

$$\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(2)}) = \frac{1}{\mathbf{s}^T \left(\mathbf{C}_j^{(1)}\right)^{-1} \mathbf{s}} = \frac{\exp(b^2) - 1}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(1)})^2} \quad (3.49)$$

and the optimal weights are

$$a_{ij,\text{opt}}^{(1)} = \frac{1/(\delta_{ij}^{(1)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(1)})^2} \tag{3.50}$$

for $i = 1, \ldots, m+n$, and $j = 1, \ldots, m$. To form the information needed for the next iteration, the blindfolded node j first generates

$$\eta_j^{(1)} = \sum_{l=1}^{m+n} \frac{1}{(\delta_{lj}^{(1)})^2}, \tag{3.51}$$

and then

$$\eta_{ji}^{(1)} = \eta_j^{(1)} - \frac{1}{(\delta_{ij}^{(1)})^2}. \tag{3.52}$$

Finally,

$$\gamma_{ji}^{(1)} = \frac{1}{\eta_{ji}^{(1)}}. \tag{3.53}$$

It should be noticed that the cooperative linear combination estimator in the previous paper [56] stops at the analytical part (knowing perfect directions) in the 1-st stage. It assigns the penalty only with the anchor's information. The new version deals with the network of few connections better than the previous one, although they have similar results in highly connected situation. In addition, since the previous method extracts no incoming information; it may have convergence problems and lower accuracy.

In the 2-nd stage, a blindfolded node might have received information from other blindfolded nodes. When this blindfolded node sends information to another node, it should remove the information from another node which previously provides estimates to itself. Hence the provider's estimated location $\hat{\omega}_{ij}^{(2)}$ should follow (3.14). Moreover, the estimate providers ($\hat{\theta}_{i,\text{opt}}^{(2)}$'s) of node j could be correlated. As a short

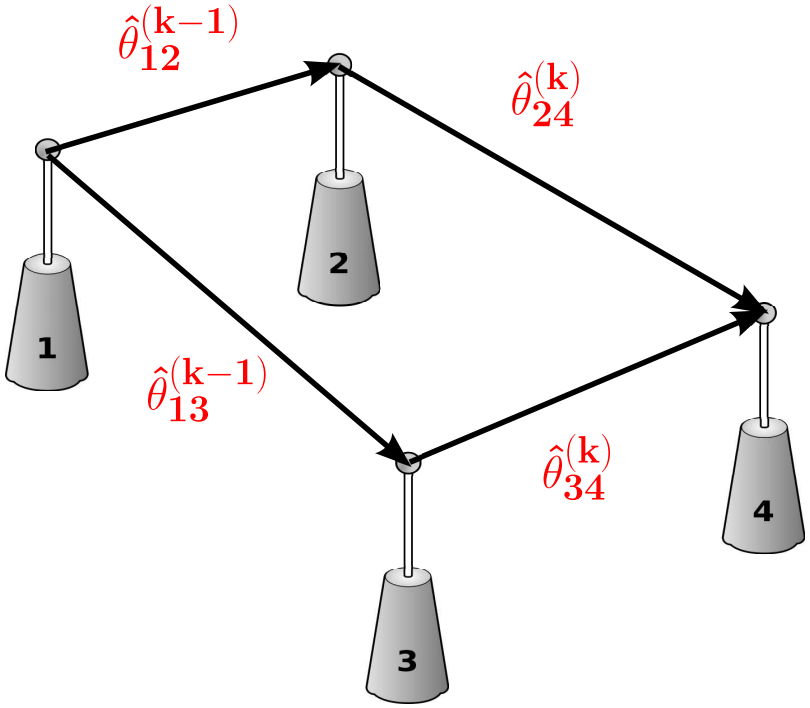

Figure 3.1: An example of correlated location estimates: $\hat{\theta}_{24}^{(k)}$ and $\hat{\theta}_{34}^{(k)}$.

cycle illustrated in Figure 3.1, node 4 has two estimate providers: node 2 and node 3. They are correlated since they both have node 1's information. However, all estimates are still assumed to be uncorrelated in order to assign weights and combine estimates easily. The results corresponding to node j if estimates are uncorrelated are derived as follows:

$$\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(3)}) = \frac{\exp(b^2)-1}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(2)})^2}; \tag{3.54}$$

$$a_{ij,\mathrm{opt}}^{(2)} = \frac{1/(\delta_{ij}^{(2)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(2)})^2} \tag{3.55}$$

where

$$\delta_{ij}^{(2)} = \sqrt{\gamma_{ij}^{(1)} + \frac{d_{ij}^2}{h_{ij}}} = \sqrt{\frac{1}{\sum_{l=1,l\neq j}^{m+n} \frac{1}{(\delta_{li}^{(1)})^2}} + \frac{d_{ij}^2}{h_{ij}}}. \tag{3.56}$$

Similarly to the previous stages, the information for the 3-rd stage is

$$\eta_j^{(2)} = \sum_{l=1}^{m+n} \frac{1}{(\delta_{lj}^{(2)})^2}. \tag{3.57}$$

Then

$$\eta_{ji}^{(2)} = \eta_j^{(2)} - \frac{1}{(\delta_{ij}^{(2)})^2}, \tag{3.58}$$

and

$$\gamma_{ji}^{(2)} = \frac{1}{\eta_{ji}^{(2)}}. \tag{3.59}$$

In summary, the iterative algorithm starts from the blindfolded nodes which have direct links to anchors. The estimated nodes then keep propagating their estimated locations to their receivers to derive an estimate or a better estimate. The similar iterative procedures are applied until $\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k+1)}) \approx \mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k)})$ for all j. This means that all information has been passed to all blindfolded nodes. For example, a full connectivity network should stop the iteration after the 2-nd stage. However, this topology may requires more steps because the dependence is not completely removed. Furthermore, the derived optimal variance (also MSE) is still less than or equal to the MSE using the perfect directions and optimal weights because the small dependence could exist even after a sender removes the information from the receivers. On the other hand, if the coming information from the estimated node is not extracted, the algorithm may not converge. Moreover, the related "optimal" MSE without extraction is incorrect and it is lower than the one with extraction.

To conclude this subsection, twos analytical parts of cooperative linear combination estimator are presented in iterative format. The first one provides estimated locations and the second one focuses on the optimal MSE.

The estimated location under perfect directions and optimal weights for each

measurement can be executed as follows.

Firstly, each anchor i $(i = m+1, \ldots, m+n)$ sets its penalties:

$$\hat{\theta}_i^{(k)} = \hat{\omega}_{ij}^{(k)} = \theta_i, \tag{3.60}$$

$$\gamma_{ij}^{(k)} = 0 \tag{3.61}$$

for all k and $j = 1, \ldots, m$.

The algorithm starts for each blindfolded node j $(j = 1, \ldots, m)$:

1. Set the connection indicator h_{ij} as in (3.3):

$$h_{ij} = \begin{cases} 1 & \text{if} \quad P_{ij} > p_{\text{th}} \quad \text{and} \quad i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3.62}$$

   for $i = 1, \ldots, m+n$.

2. Calculate unbiased range estimates $\hat{d}_{ij}$ from its received powers:

$$\hat{d}_{ij} = \exp(-0.5b^2)\frac{d_0}{h_{ij}}\left(\frac{p_{ij}}{p_0}\right)^{-\frac{1}{n_p}} \quad \text{for} \quad i = 1, \ldots, m+n. \tag{3.63}$$

3. Compute the true normalized direction from node $i = 1, \ldots, m+n$ to node j

$$\mathbf{v}_{ij} = \begin{cases} \frac{\theta_j - \theta_i}{\|\theta_j - \theta_i\|} & \text{if} \quad \|\theta_j - \theta_i\| \neq \mathbf{0} \\ \mathbf{0} & \text{otherwise} \end{cases}. \tag{3.64}$$

   The $\|\cdot\|$ represents the Euclidean norm.

4. Assign the initial penalties and location estimate of blindfolded node j:

$$\gamma_{ji}^{(-1)} = \infty, \tag{3.65}$$

$$\hat{\omega}_{ji}^{(0)} = \text{any vector}, \tag{3.66}$$

respectively, for $i = 1, \ldots, m$.

5. Set $k = 0$ to start the following iteration after all blindfolded nodes finish all previous steps.

6. Calculate the square of the effective distances

$$(\delta_{ij}^{(k)})^2 = \gamma_{ij}^{(k-1)} + \frac{d_{ij}^2}{h_{ij}} \quad \text{for} \quad i = 1, \ldots, m+n. \tag{3.67}$$

7. Compute node j's penalties:

$$\eta_j^{(k)} = \sum_{l=1}^{m+n} \frac{1}{(\delta_{lj}^{(k)})^2}, \tag{3.68}$$

$$\eta_{ji}^{(k)} = \eta_j^{(k)} - \frac{1}{(\delta_{ij}^{(k)})^2}, \tag{3.69}$$

$$\gamma_{ji}^{(k)} = \frac{1}{\eta_{ji}^{(k)}}. \tag{3.70}$$

for $i = 1, \ldots, m$.

8. The optimal weights of node j is given by

$$a_{ij,\text{opt}}^{(k)} = \frac{1/(\delta_{ij}^{(k)})^2}{\eta_j^{(k)}} = \frac{1/(\delta_{ij}^{(k)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(k)})^2} \quad \text{for} \quad i = 1, \ldots, m+n. \tag{3.71}$$

9. The new location estimate for jth node is given by

$$\hat{\theta}_j^{(k+1)} = \sum_{i=1}^{m+n} a_{ij,\text{opt}}^{(k)} \hat{\theta}_{ij}^{(k)} = \sum_{i=1}^{m+n} a_{ij,\text{opt}}^{(k)} \left(\hat{\omega}_{ij}^{(k)} + \hat{d}_{ij} \mathbf{v}_{ij} \right) \tag{3.72}$$

and the new extracted location estimate

$$\hat{\omega}_{ji}^{(k+1)} = \frac{\hat{\theta}_j^{(k+1)} - a_{ij,\text{opt}}^{(k)} (\hat{\omega}_{ij}^{(k)} + \hat{d}_{ij} \mathbf{v}_{ij})}{1 - a_{ij,\text{opt}}^{(k)}} \quad \text{for} \quad i = 1, \ldots, m. \tag{3.73}$$

10. Stop the procedure if

$$\frac{S^{(k)}}{N^{(k)}} \equiv \frac{\sum_{j=1}^{m} S_j^{(k)}}{\sum_{j=1}^{m} N_j^{(k)}} \equiv \frac{\sum_{j=1}^{m} \left\| \hat{\theta}_j^{(k+1)} - \hat{\theta}_j^{(k)} \right\|^2}{\sum_{j=1}^{m} \left\| \hat{\theta}_j^{(k)} \right\|^2} < \epsilon \tag{3.74}$$

or

$$k \geq \text{maximum number of iterations.} \tag{3.75}$$

Otherwise, let $k = k + 1$ and node j sends $\hat{\omega}_{ji}^{(k+1)}$, $\hat{\gamma}_{ji}^{(k)}$, $S^{(k)}$, and $N^{(k)}$, and to the network and the data will be received by the node i if $h_{ji} = 1$. Then repeat steps 6) $-$ 10).

$S^{(k)}$ and $N^{(k)}$ represent the k-th stage of global variables S and N which are updated by $S - S_j^{(k-1)} + S_j^{(k)}$ and $N - N_j^{(k-1)} + N_j^{(k)}$ at node j, respectively.

Running the above procedure for many power measurements and averaging the estimation error can provide the optimal simulated MSE. The easy and fast method to obtain the optimal MSE (variance) given a topology without many trails is presented. It should be noticed that the optimal MSE is slightly lower than the simulated MSE in the previous paragraphs because the small dependence remains.

First of all, each anchor i $(i = m+1, \ldots, m+n)$ sets its penalties:

$$\gamma_{ij}^{(k)} = 0 \quad \text{for all } k \text{ and } j = 1, \ldots, m. \tag{3.76}$$

The algorithm starts for each blindfolded node j $(j = 1, \ldots, m)$:

1. Set the connection indicator h_{ij} as in (3.3):

$$h_{ij} = \begin{cases} 1 & \text{if } \; P_{ij} > p_{\text{th}} \; \text{ and } \; i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3.77}$$

   for $i = 1, \ldots, m+n$.

2. Assign the initial penalties of blindfolded node j:

$$\gamma_{ji}^{(-1)} = \infty \quad \text{for} \quad i = 1, \ldots, m. \tag{3.78}$$

3. Set $k = 0$ to start the following iteration after all blindfolded nodes finish all previous steps.

4. Calculate the square of the effective distances

$$(\delta_{ij}^{(k)})^2 = \gamma_{ij}^{(k-1)} + \frac{d_{ij}^2}{h_{ij}} \quad \text{for} \quad i = 1, \ldots, m. \tag{3.79}$$

5. Compute node j's penalties:

$$\eta_j^{(k)} = \sum_{l=1}^{m+n} \frac{1}{(\delta_{lj}^{(k)})^2}, \tag{3.80}$$

$$\eta_{ji}^{(k)} = \eta_j^{(k)} - \frac{1}{(\delta_{ij}^{(k)})^2}, \tag{3.81}$$

$$\gamma_{ji}^{(k)} = \frac{1}{\eta_{ji}^{(k)}}. \tag{3.82}$$

for $i = 1, \ldots, m$.

6. The optimal MSE of node j is given by

$$\mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k+1)}) = \frac{\exp(b^2) - 1}{\eta_j^{(k)}} = \frac{\exp(b^2) - 1}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(k)})^2} \tag{3.83}$$

7. Stop the procedure if

$$\frac{S^{(k)}}{N^{(k)}} \equiv \frac{\sum_{j=1}^{m} S_j^{(k)}}{\sum_{j=1}^{m} N_j^{(k)}} \equiv \frac{\sum_{j=1}^{m} \left\| \mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k+1)}) - \mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k)}) \right\|^2}{\sum_{j=1}^{m} \left\| \mathrm{Var}(\hat{\theta}_{j,\mathrm{opt}}^{(k)}) \right\|^2} < \epsilon \tag{3.84}$$

or

$$k \geq \text{maximum number of iterations.} \tag{3.85}$$

Otherwise, let $k = k + 1$ and node j sends $S^{(k)}$, $N^{(k)}$, and $\gamma_{ji}^{(k)}$ to the network and the data will be received by the node i if $h_{ji} = 1$. Then repeat steps 4) − 7).

$S^{(k)}$ and $N^{(k)}$ represent the k-th stage of global variables S and N which are updated by $S - S_j^{(k-1)} + S_j^{(k)}$ and $N - N_j^{(k-1)} + N_j^{(k)}$ at node j, respectively.

If the network is fully connected, the maximum number of iterations can be 2.

3.3.2 Linear Combination Estimator in Reality

This subsection presents the detailed steps of the linear combination estimator in a practical environment where the directions also need to be estimated. The optimal weights and true directions which depend on the true locations are impossible to know in reality. First of all, the (true) effective distances resulting in combining weights are replaced by the estimated effective distances and the directions are obtained from previous estimated locations. Secondly, the procedure of extracting all information from receivers as shown in the analysis of the LC estimator in the previous subsection is not necessary because the extraction slightly improves the accuracy while increasing overhead. If the network is in full connectivity, a blindfolded node is required to generate $m-1$ different estimates and the corresponding $m-1$ uncertainties (penalties)+ of its location in each iteration. The overhead of extracting information could be $m-1$ times more than the one without extracting. Moreover, removing the dependence is difficult while directions are also estimated.

Therefore, the two following LC algorithms with slight differences are given: 1) LC without extraction (abbr. LC) and 2) LC using weights with extraction and location estimates without extraction (abbr. LCa). The reason to execute 2) is that it reduces iterations which provides less inter-node communications while the overhead is acceptable. For the LC without extraction, a blindfolded node i using a LC estimator sends the same location estimate to all its receivers without customizing for each receiver, i.e., uses $\hat{\theta}_i^{(k)}$ instead of $\hat{\omega}_{ij}^{(k)}$, and the effective distance including the information from the receiver by using $\hat{\gamma}_i^{(k-1)}$ instead of $\hat{\gamma}_{ij}^{(k-1)}$. The LC using weights with extraction means to use the $\hat{\gamma}_{ij}^{(k-1)}$ for effective distance and $\hat{\theta}_i^{(k)}$ for the provider's location.

In summary, the LC algorithm in reality iteratively and simultaneously refine

locations and corresponding directions until the difference between previous estimated and current locations is small enough or reaches the given maximal number of iterations.

The details of the LC algorithms without extraction and with weight extraction are given in the following paragraphs. The $k = 0, 1, 2, \ldots$ will represent the step of iterations; blindfolded nodes are labeled as $1, 2, \ldots, m$ and anchors are as $m+1, m+2, \ldots, m+n$.

First of all, each anchor i ($i = m+1, \ldots, m+n$) sets its location (estimate) and penalties:

$$\hat{\theta}_i^{(k)} = \theta_i, \tag{3.86}$$

$$\hat{\gamma}_i^{(k)} = \hat{\gamma}_{ij}^{(k)} = 0, \tag{3.87}$$

respectively, for all k and $j = 1, \ldots, m$.

The algorithm starts for each blindfolded node j ($j = 1, \ldots, m$):

1. Set the connection indicator h_{ij} as in (3.3):

$$h_{ij} = \begin{cases} 1 & \text{if } \; P_{ij} > p_{\text{th}} \; \text{ and } \; i \neq j \\ 0 & \text{otherwise} \end{cases} \tag{3.88}$$

   for $i = 1, \ldots, m+n$.

2. Calculates unbiased range estimates $\hat{d}_{ij}$ from its received powers; it is the same as (2.10) except for having h_{ij} in the denominator:

$$\hat{d}_{ij} = \exp(-0.5b^2)\frac{d_0}{h_{ij}}\left(\frac{p_{ij}}{p_0}\right)^{-\frac{1}{n_p}} \quad \text{for} \quad i = 1, \ldots, m+n. \tag{3.89}$$

Obviously, $\hat{d}_{ij} = \infty$ if the connection fails to build ($h_{ij} = 0$).

3. Use $\hat{\theta}_i^{(0)}$ and $\hat{d}_{ij}$ for $i = m+1, \ldots, m+n$ ($\hat{\delta}_{ij}^{(0)}$) through the non-cooperative LC in Section 2.3 to provide the initial estimates $\hat{\theta}_j^{(1)}$. If no anchors connect to node i, assign $\hat{\theta}_j^{(1)}$ = any vector because the corresponding weights will be 0.

   Also compute node j's penalties:

$$\hat{\gamma}_j^{(0)} = \hat{\gamma}_{ji}^{(0)} = \frac{1}{\sum_{l=m+1}^{m+n} \frac{h_{lj}}{\hat{d}_{lj}^2}} \quad \text{for} \quad i = 1, \ldots, m. \tag{3.90}$$

4. Set $k = 1$ to start the following iteration after all blindfolded nodes obtains their initial estimates and penalties.

5. Compute the normalized direction from node $i = 1, \ldots, m+n$ to node j in the k-th stage:

$$\mathbf{v}_{ij}^{(k)} = \begin{cases} \frac{\hat{\theta}_j^{(k)} - \hat{\theta}_i^{(k)}}{\left\| \hat{\theta}_j^{(k)} - \hat{\theta}_i^{(k)} \right\|} & \text{if} \quad \left\| \hat{\theta}_j^{(k)} - \hat{\theta}_i^{(k)} \right\| \neq \mathbf{0} \\ \mathbf{0} & \text{otherwise} \end{cases} . \tag{3.91}$$

   The $\| \cdot \|$ represents the Euclidean norm.

6. Compute the square of the estimated effective distances by one of the following:

   (a) Without extraction:

$$(\hat{\delta}_{ij}^{(k)})^2 = \hat{\gamma}_i^{(k-1)} + \frac{\left\| \hat{\theta}_j^{(k)} - \hat{\theta}_i^{(k)} \right\|^2}{h_{ij}} \tag{3.92}$$

   or

(b) With weight extraction:

$$(\hat{\delta}_{ij}^{(k)})^2 = \hat{\gamma}_{ij}^{(k-1)} + \frac{\left\| \hat{\theta}_j^{(k)} - \hat{\theta}_i^{(k)} \right\|^2}{h_{ij}} \tag{3.93}$$

for $i = 1, \ldots, m+n$.

7. Compute the denominator for combining weights

$$\hat{\eta}_j^{(k)} \equiv \sum_{l=1}^{m+n} \frac{1}{(\hat{\delta}_{lj}^{(k)})^2}, \tag{3.94}$$

its reciprocal

$$\hat{\gamma}_j^{(k)} \equiv \frac{1}{\hat{\eta}_j^{(k)}} = \frac{1}{\sum_{l=1}^{m+n} 1/(\hat{\delta}_{lj}^{(k)})^2}, \tag{3.95}$$

and the combining weights

$$a_{ij}^{(k)} = \frac{1/(\hat{\delta}_{ij}^{(k)})^2}{\sum_{l=1}^{m+n} 1/(\hat{\delta}_{lj}^{(k)})^2} \equiv \frac{\hat{\gamma}_j^{(k)}}{(\hat{\delta}_{ij}^{(k)})^2} \tag{3.96}$$

for $i = 1, \ldots, m+n$.

8. The new location estimate for node j is given by

$$\hat{\theta}_j^{(k+1)} = \sum_{i=1}^{m+n} a_{ij}^{(k)} \hat{\theta}_{ij}^{(k)} = \sum_{i=1}^{m+n} a_{ij}^{(k)} \left(\hat{\theta}_i^{(k)} + \hat{d}_{ij} \mathbf{v}_{ij}^{(k)} \right) \tag{3.97}$$

9. If use the LC with weight extraction, compute

$$\hat{\eta}_{ji}^{(k)} = \hat{\eta}_j^{(k)} - \frac{1}{(\hat{\delta}_{ij}^{(k)})^2}, \tag{3.98}$$

and

$$\hat{\gamma}_{ji}^{(k)} \equiv \frac{1}{\hat{\eta}_{ji}^{(k)}} \tag{3.99}$$

for $i = 1, \ldots, m$.

10. Stop the procedure if

$$\frac{S^{(k)}}{N^{(k)}} \equiv \frac{\sum_{j=1}^{m} S_j^{(k)}}{\sum_{j=1}^{m} N_j^{(k)}} \equiv \frac{\sum_{j=1}^{m} \left\| \hat{\theta}_j^{(k+1)} - \hat{\theta}_j^{(k)} \right\|^2}{\sum_{j=1}^{m} \left\| \hat{\theta}_j^{(k)} \right\|^2} < \epsilon \tag{3.100}$$

or

$$k \geq \text{maximum number of iterations.} \tag{3.101}$$

Otherwise, let $k = k + 1$ and node j sends $\hat{\theta}_j^{(k+1)}$, $S^{(k)}$, $N^{(k)}$, and $\hat{\gamma}_j^{(k)}$ (or $\hat{\gamma}_{ji}^{(k)}$ for weight extraction LC) to the network and the data will be received by the node i if $h_{ji} = 1$. Then repeat steps 5) $-$ 10).

$S^{(k)}$ and $N^{(k)}$ represent the k-th stage of global variables S and N which are updated by $S - S_j^{(k-1)} + S_j^{(k)}$ and $N - N_j^{(k-1)} + N_j^{(k)}$ at node j, respectively.

The difference between LC and LCa is step 6) and LCa needs to generate different uncertainties of one estimated location to other nodes in step 9). Moreover, if the LC is performed in two dimensions, $\hat{\theta}_j^{(k)}$ can be represented by $\left[\hat{x}_j^{(k)}, \hat{y}_j^{(k)}\right]$.

As mentioned before, the non-cooperative LC estimator can be viewed as the modification of cooperative LC estimator by only using the information from anchors. Then the effective distances are equal to infinity for blindfolded nodes and the unbiased range estimates for anchors in every iteration:

$$\hat{\delta}_{ij}^{(k)} = \begin{cases} \infty & \text{for} \quad i = 1, \ldots, m \\ \hat{d}_{ij} & \text{for} \quad i = m+1, \ldots, m+n \end{cases} . \tag{3.102}$$

In order to reduce the number of communications between nodes, two modifications called *full calibration* (fc) and *partial calibration* (pc) are introduced. For the full calibration, each blindfolded node (j) has an inner loop by replacing superscript (k) corresponding to j with (k_q) and repeating step 5) $-$ 8) until

$$\frac{\left\|\hat{\theta}_j^{(k_{q+1})} - \hat{\theta}_j^{(k_q)}\right\|^2}{\left\|\hat{\theta}_j^{(k_q)}\right\|^2} < \epsilon_j \tag{3.103}$$

or

$$q \geq \text{maximum number of iterations.} \tag{3.104}$$

where the subscript q represents the q-th iteration of the inner loop. Clearly, $\mathbf{v}_{ij}^{(k_q)}$, $\hat{\delta}_{ij}^{(k_q)}$, $\hat{\gamma}_j^{(k_q)}$, $a_{ij}^{(k_q)}$ and $\hat{\theta}_j^{(k_{q+1})}$ change with the inner iteration while keeping $\hat{\theta}_i^{(k)}$ and $\hat{\gamma}_{ij}^{(k-1)}$ (or $\hat{\gamma}_i^{(k-1)}$) unchanged. After node j finishes its inner iteration, it substitutes $\hat{\theta}_j^{(k+1)}$ and $\hat{\gamma}_j^{(k)}$ (or $\hat{\gamma}_{ji}^{(k)}$) by $\hat{\theta}_j^{(k_{q+1})}$ and $\hat{\gamma}_j^{(k_q)}$ (or $\hat{\gamma}_{ji}^{(k_q)}$), respectively, and sends them to other nodes' information for the next iteration.

On the other hand, partial calibration refines one node's location by iteratively computing the related directions i.e., steps 5) and 9) ($\mathbf{v}_{ij}^{(k_q)}$ and $\hat{\theta}_j^{(k_{q+1})}$). However, the estimated effective distances and resulting weights remain unchanged until every node finishes its calibration. This partial calibration part is similar to non-cooperative LC where the transmitters' locations and combining weights are unchanged and only the directions are improved.

Another modification is to use the most current information from the other nodes like the coordinate descent method and Gsuss-Seidel algorithm. Explicitly, in the k-th step, each node *sequentially* (instead of performing a parallel computation as before) performs the LC according to its label. Then node j uses any information from node i's $(k+1)$-th results if $i < j$. On the other hand, if $i > j$, node j has to

use node i's k-th results. This method is denoted by the abbreviation "LCn," and both time synchronization and labeling nodes become essential issues.

3.4 Examples and Simulation Results

Three mainly distributed methods are investigated by simulation in this section: the maximum likelihood estimator (MLE) by the coordinate descent method (CD), linear combination (LC), and distributed multidimensional scaling (dwMDS) [26]. Basically, the dwMDS does a linear combination, but its weight assignments are different from the LC. Other methods such as combing weight without penalty in LC, equal weight LC (the same as the sequential greedy optimization (SGO) [32]), and distributed spatially constrained localization (DSCL) [33] are also compared in few examples.

Two performance measures will be investigated: number of iterations and mean squared error (MSE). The number iterations represent the inter-node communications in a network because the radio resources, such as bandwidth and transmitting power, are precious. Less communication saves not only energy but also spectrum in wireless communications. The MSE quanties the accuracy of localization methods and the MSE for a fixed unknown location θ_i is defined as

$$\mathrm{MSE}(\hat{\theta}_i) \equiv \mathbb{E}\left[\left\|\hat{\theta}_i - \theta_i\right\|^2\right] \tag{3.105}$$

where the expectation is taken over all $\hat{\theta}_i$ and the RMSE is the square root of the MSE. In the simulation, sample RMSE is used and given by

$$\text{(Sample) RMSE}(\hat{\theta}_i) \equiv \sqrt{\frac{1}{N_{\mathrm{sim}}}\sum_{k=1}^{N_{\mathrm{sim}}}\left\|\hat{\theta}_i[k] - \theta_i\right\|^2} \tag{3.106}$$

where $\hat{\theta}_i[k]$ is the estimation result of k-th trial for node i and N_{sim} is the total number of trials. The word "sample" will be omitted for simplicity when showing the simulation results. For m fixed blindfolded nodes, the average (sampel) RMSE is given by

$$\text{Avg. (Sample) RMSE}(\hat{\boldsymbol{\theta}}) \equiv \frac{1}{m}\sum_{i=1}^{m}\sqrt{\frac{1}{N_{\text{sim}}}\sum_{k=1}^{N_{\text{sim}}}\left\|\hat{\theta}_i[k]-\theta_i\right\|^2} \tag{3.107}$$

where $\hat{\boldsymbol{\theta}} \equiv [\hat{\theta}_1, \ldots, \hat{\theta}_m]$. On the other hand, if the topology in the network is random, the MSE of a randomly placed blindfolded node Θ_i should average over both $\hat{\Theta}_i$ and Θ_i, i.e.,

$$\text{MSE}(\hat{\Theta}_i) \equiv \mathbb{E}\left[\left\|\hat{\Theta}_i-\Theta_i\right\|^2\right] = \mathbb{E}_{\Theta_i}\left[\mathbb{E}_{\hat{\Theta}_i}\left[\left\|\hat{\Theta}_i-\Theta_i\right\|^2\middle|\Theta_i\right]\right] \tag{3.108}$$

However, during the simulations, each random topologies only run one set of random power measurement and the sample MSE can be written as

$$\text{(Sample) MSE}(\hat{\Theta}_i) \equiv \frac{1}{N_{\text{sim}}}\sum_{k=1}^{N_{\text{sim}}}\left\|\hat{\theta}_i[k]-\theta_i[k]\right\|^2. \tag{3.109}$$

Similarly, the sample MSE for the network with m blindfolded nodes is

$$\text{(Sample) MSE}(\hat{\boldsymbol{\Theta}}) \equiv \frac{1}{N_{\text{sim}}}\sum_{k=1}^{N_{\text{sim}}}\frac{1}{m}\sum_{i=1}^{m}\left\|\hat{\theta}_i[k]-\theta_i[k]\right\|^2 \tag{3.110}$$

where $\hat{\boldsymbol{\Theta}} \equiv [\hat{\Theta}_1, \ldots, \hat{\Theta}_m]$. The sample RMSE then is given by taking square root of sample MSE

$$\text{(Sample) RMSE}(\hat{\boldsymbol{\Theta}}) \equiv \sqrt{\frac{1}{N_{\text{sim}}}\sum_{k=1}^{N_{\text{sim}}}\frac{1}{m}\sum_{i=1}^{m}\left\|\hat{\theta}_i[k]-\theta_i[k]\right\|^2} \tag{3.111}$$

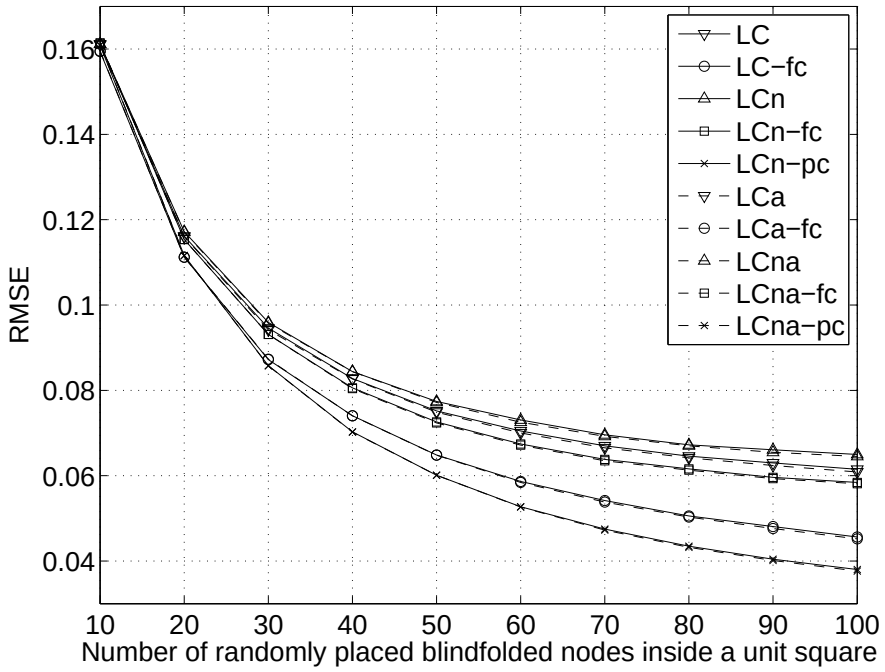

Figure 3.2: RMSE of various LCs versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

Again, "sample" will be omitted when showing the simulation results. Because the convergence of LC is uncertain, the percentage of convergence (having a fixed point) will be shown.

3.4.1 Full Connectivity inside a Unit Square

The first example uses the same topology with four anchors at the corners in a unit square as in non-cooperative localization. The blindfolded nodes are randomly placed in the observation region according to a uniform distribution. The order of nodes affect the MLE:CD and LCn using successively update. However, the ordering is randomly assigned without elaborate design. All nodes in the network are assumed to directly connect to all other nodes, i.e., full connectivity. Because of the random topologies, the sample RMSE (3.111) is applied. In the MLE: CD, the same step 0.005

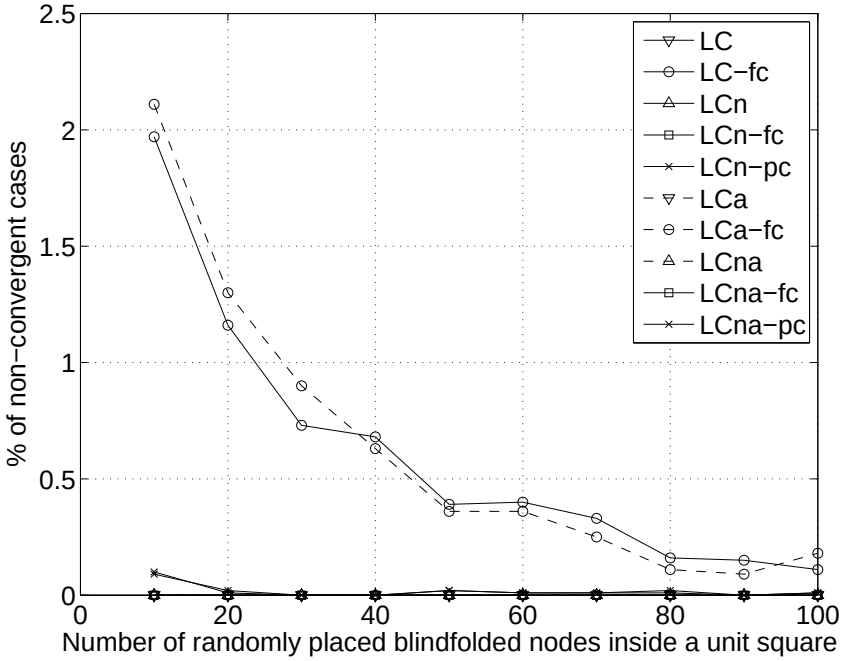

Figure 3.3: Non-convergence percentages of various LC estimators versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

is applied for both search spaces which makes $\mathcal{S}_{x_i} = \mathcal{S}_{y_i} = \{0, 0.005, \ldots, 0.995, 1\}$. Thus CD finds the minimum from 200 candidates in each coordinate and CD has bounded estimates which are advantage over LC and dwMDS. The initial estimates are also computed by CD in non-cooperative sense by starting at a random guess inside the unit square. The initial CD uses the stopping $\epsilon = 10^{-6}$ in (2.37) and 100 as the permitted maximum number of iterations for each blindfolded node. The cooperative MLE:CD use the same stopping criteria as LC in (3.100) and ϵ set to be 10^{-8}, and the permitted maximum number of iterations is 10^4. The same ϵ and the allowed number of iterations are also used in LC where the non-cooperative LC provides the initial estimates $\hat{\theta}_i^{(1)}$ for $i = 1, \ldots, m$ with the stopping $\epsilon = 10^{-6}$ and 100 iterations at most. If LC would like to apply full calibration (fc) or partial cal-

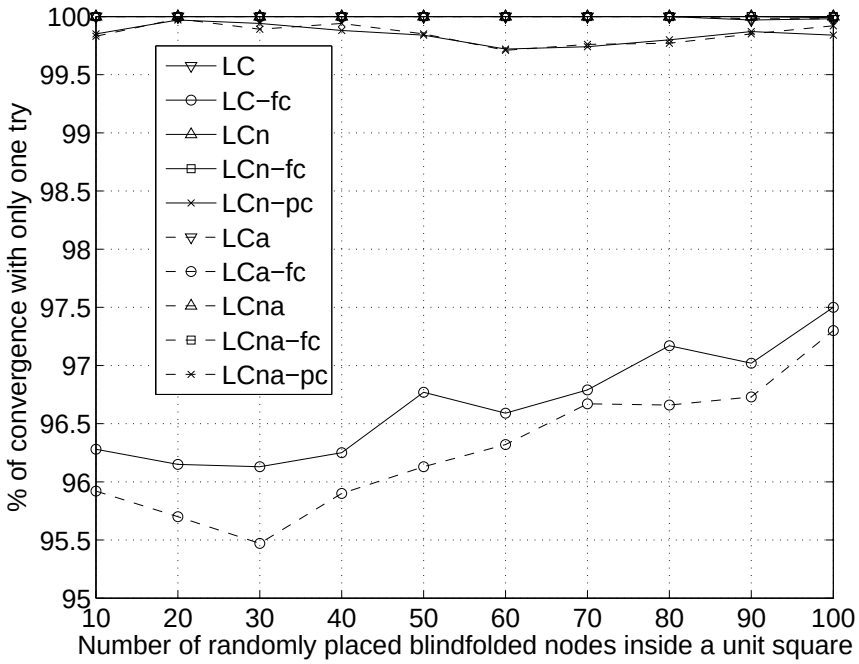

Figure 3.4: Percetange of various LCs that converge with one run versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

ibration (pc), it uses the same stopping parameters as the noon-cooperative LC. If a result of LCs is not convergent, the LCs allows all nodes in the network to have another try by using another random guess inside the unit square to compute the initial estimate. The maximal number of reruns is set to be 10. The dwMDS finishes when the difference of current STRESS function and previous one is small enough, i.e., $S^{(k-1)} - S^{(k)} < \epsilon$. A small modification of stopping dwMDS is given by

$$\frac{|S^{(k-1)} - S^{(k)}|}{S^{(k-1)}} < \epsilon \tag{3.112}$$

This stopping criterion is similar to that of LC and let $\epsilon = 10^{-8}$. The dwMDS can also use the criterion in LC, and the two corresponding RMSEs are very similar. Again, the iterations of dwMDS is at most 10^4. According to my simulations, dwMDS

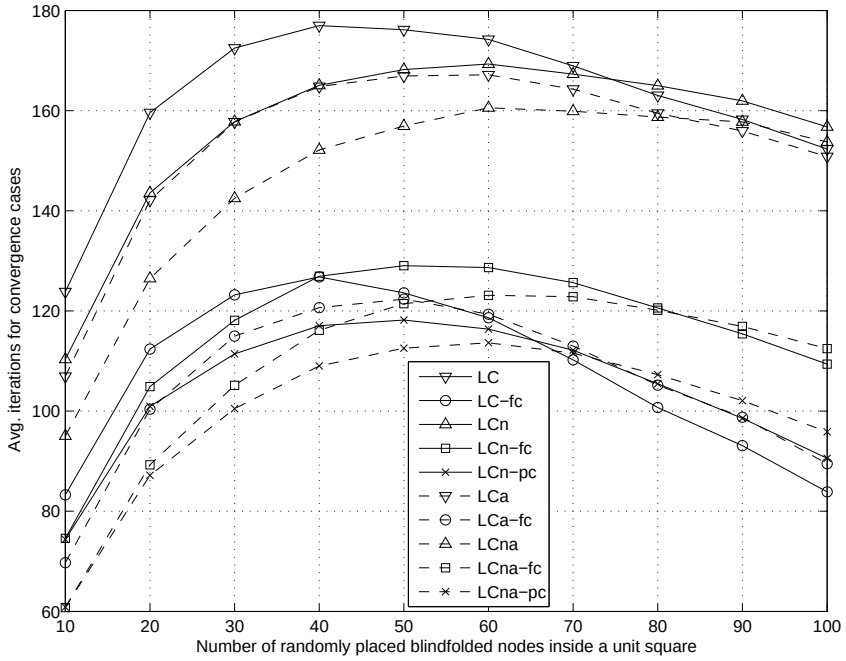

Figure 3.5: Average iterations of various LCs versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

requires the accuracy of initial guess within a certain range. Thus, the results of non-cooperative LC serve as the initial points for dwMDS.

As mentioned before, there are various modifications of LC algorithms. Figure 3.2 shows the RMSE of various LCs versus the number of nodes under full connectivity with $n = 3$ and $\sigma = 5$. The original LC has the second worst RMSE and LCn where nodes use other's updated position serially instead of parallely as LC has the worst RMSE. For the full calibration (fc) case, the parallel one (LC-fc) has better accuracy than the serial one (LCn-fc). Serial partial calibration (LCn-pc) has the best RMSE=0.038 among 5 LCs while 100 blindfolded nodes. The other five versions which use weights with extraction are denoted with "a" and shown in dash lines. It is found that RMSEs with and without extraction are almost identical. The reason

for serial update performing worse than parallel update might be early refining of uncertain estimates. Similar phenomena happens to fc and pc. Elaborately refining combining weights from uncertain estimates could worsen the result. Because the convergence of LCs is not guaranteed, simulations are applied to check how often convergence happens with given stopping $\epsilon = 10^{-8}$, maximal 10^4 iterations, and up to 10 retries in Figure 3.3. LC-fc and LCa-fc are the worst two and they have the non-convergent case from 0.09% to 2.1% of the time. For those non-convergent estimates, many of them are oscillating. i.e., $\hat{\theta}_j^{(k)} \approx \hat{\theta}_j^{(k-2)}$. One can smooth the estimate as $(\hat{\theta}_j^{(k)} + \hat{\theta}_j^{(k-1)})/2$ to make it convergent. On the other hand, all 10^4 are convergent for LC, LCa, LCn, LCn, LCn-fc, and LCna-fc. The LCn-pc has about 0.1% non-convergent cases and LCna-pc has almost all convergent cases. Figure 3.4 shows the percentages of LCs with only one try to obtain a convergent estimate. Most LCs can converge with one run except LC-fc, LCa-fc, LCn-pc, and LCna-pc. More than 99.7% of LCn-pc and LCna-pc converge without another try, and more than 95.5% of LC-fc and LCa-fc as well. If an estimator does not converge, it runs to the maximal iterations which is 10^4 in the setup. Therefore, instead of the average number of iterations, average number of iterations for "convergence" are presented and given in Figure 3.5. LC, LCn, LCa, and LCna require the most iterations and the numbers can be more than 160. LC-fc has the lowest iterations in a dense network. The number of iterations for LCn-pc and LCna-pc are lower than LC-fc and LCa-fc in a looser network. Another observation is the concave shape for the LCs. This means the number of iterations are bounded while increasing the number of blindfolded nodes.

Figure 3.6 incorporates MLE:CD, dwMDS, and three representative LCs: LC, LC-fc and LCn-pc. The dwMDS generally has the worst RMSE. The MLE:CD and LCn-pc work best in the lower density and higher density regimes, respectively. All

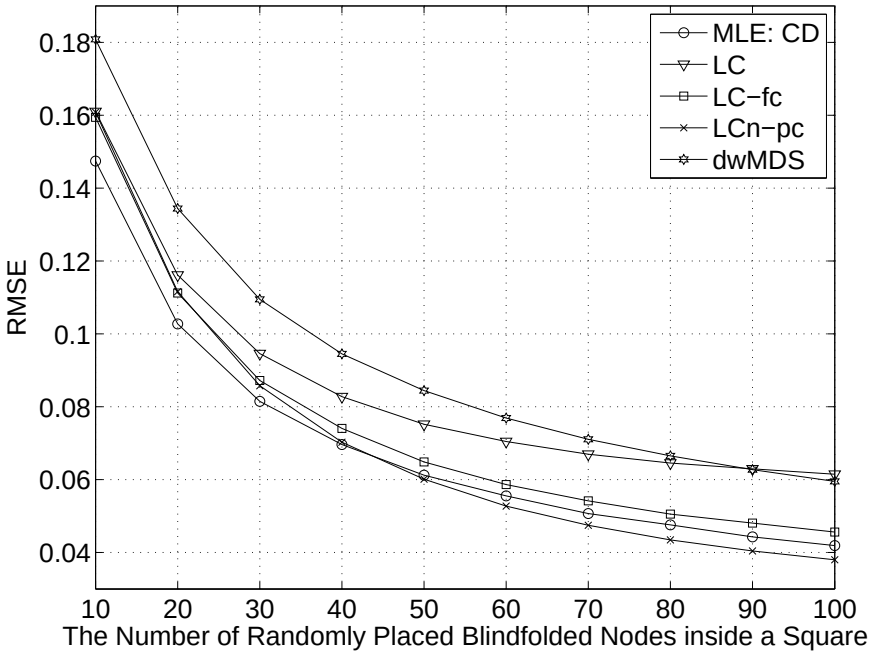

Figure 3.6: RMSE of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

simulations in both MLE:CD and dwMDS are all convergent. It is noticeable that if CD uses the parallel update, CD may not converge. The average number of iterations for the above five algorithms are shown in Figure 3.7. The MLE:CD requires fewest iterations when the search step size is 0.005. dwMDS requires second fewest iterations when fewer nodes in the network. However, the iterations of dwMDS increases when the number of nodes increases, which is a drawback.

The performance of low SNR case with $n_p = 1.5$ and $\sigma = 5$ are shown in Figures 3.8 - 3.10. First of all, the RMSEs of all methods are worse than those with $n_p = 3$. The MLE:CD performs much better than LC estimators because the combing weights of LC estimators are disturbed by worse range estimates under the low SNR regime. The dwMDS also have the similar problem but its RMSE is much

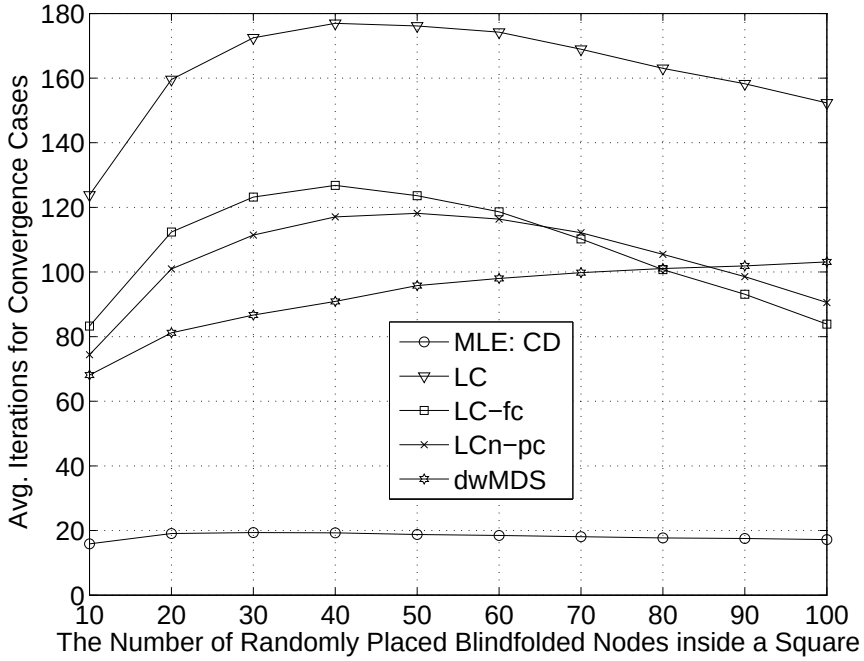

Figure 3.7: Average iterations of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

closer to LCn-pc now. The average number of iterations in the LC estimators are increasing a lot while the iterations of MLE:CD and dwMDS are slightly increasing. The non-convergent cases of LC-fc becomes larger especially in the low density case.

Figures 3.11 and 3.12 present the high SNR case with $n_p = 5$ and $\sigma = 5$. All methods have much lower RMSE and the RMSEs of the LC estimators are excellent. Even the original LC has a similar accuracy as compared with the MLE:CD. On the other hand, the curve of dwMDS is far from others which means the dwMDS takes less advantage of better range estimates. The average iterations are also decreasing and MLE:CD still requires very few inter-connections. Almost all methods converge in this situation, the only one non-convergent case out of 10^4 simulations happened with LCn-pc with 80 blindfolded nodes.

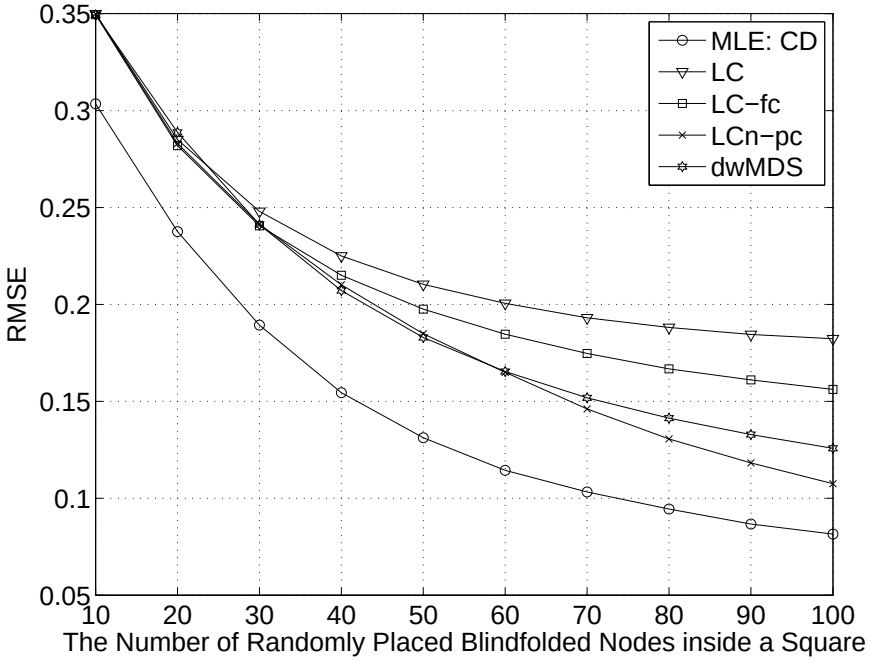

Figure 3.8: RMSE of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 1.5$ and $\sigma = 5$.

Besides considering three major compared algorithms: LC, MLE:CD, and dwMDS, other related distributed algorithms are also investigated. Instead of estimated effective distances, the LC estimator may use the unbiased estimated distances to compute the weights. This is the same as the non-cooperative LC and is denoted as 'NLC'. Another weight assignment is done by equal weight (by abbreviation EW), which means $a_{ij} = 1/(m+n)$ for totally $m+n$ nodes connected to the target blindfolded node. The equal weight with LC has the same form as the sequential greedy optimization (SGO) [32], expect its initial guess from the result of second-order cone programming (SOCP). The easiest way to compare our LC with SCOP+SGO is by letting the true locations be the initial guess for SGO. An exhaustive search based algorithm, distributed spatially constrained localization (DSCL) [33] is also presented.

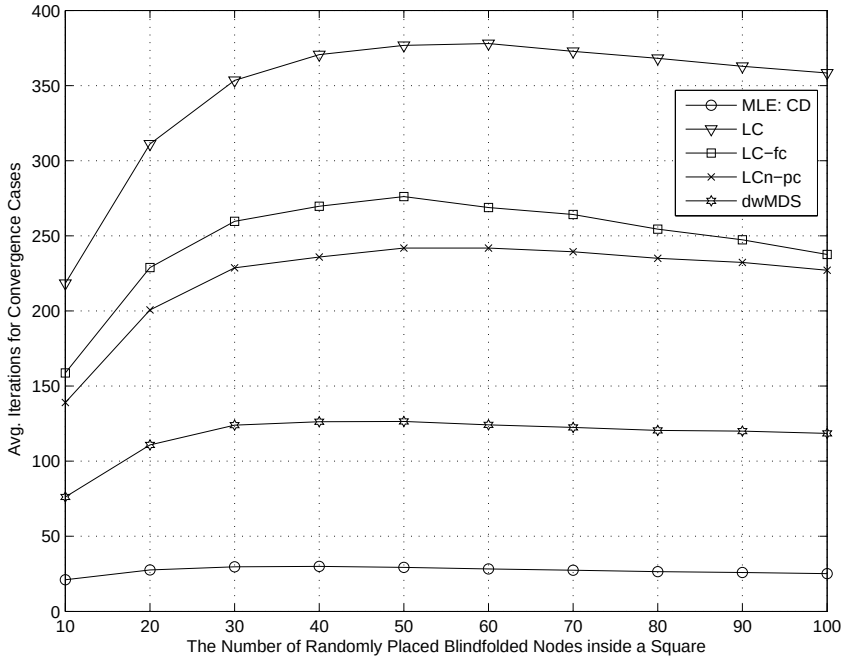

Figure 3.9: Average iterations of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 1.5$ and $\sigma = 5$.

This method computes candidates around the previous estimate and chooses the candidate with minimal cost to be the new estimate.

Figures 3.13–3.15 show the RMSE of the above methods compared with LC and dwMDS. In most cases, the NLC has the highest RMSE. However, the RMSE of NLCn-pc is slightly higher than LCn-pc and it means the partial calibration helps the accuracy. DSCL performs better in the low SNR and low density regime and its RMSE improves little when the SNR or density becomes better. The EW with true initial point, which can be viewed as the SGO with perfect initial point, has similar performance with the dwMDS. This demonstrates the accuracy of SCOP+SGO worse than dwMDS and the proposed LCn-pc. Moreover, it shows that elaborating and careful weight assignment such as the LC family helps the accuracy a lot in received

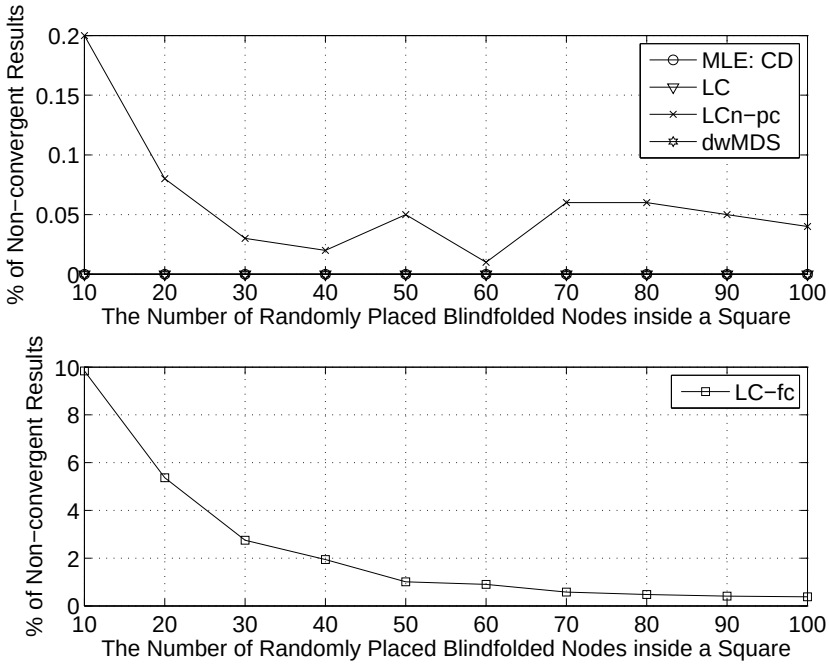

Figure 3.10: Non-convergence percentages of various algorithms versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 1.5$ and $\sigma = 5$.

signal model.

3.4.2 Partial Connectivity inside a Unit Square

It is necessary to consider a partial connectivity instead of full connectivity because transmitting power of a node usually cannot cover the whole network, especially in a sensor network. A unit square is still the observation area and 4 anchors are at corner. Blindfolded nodes are randomly placed inside the unit square. The above setup is the same as the fully connected case. The connection between nodes i and j is built if $P_{ij} > p_{\text{th}}$ as in (3.3) and

$$p_{\text{th}} = p_0 \left(\frac{d_{\text{th}}}{d_0} \right)^{-n_p} \tag{3.113}$$

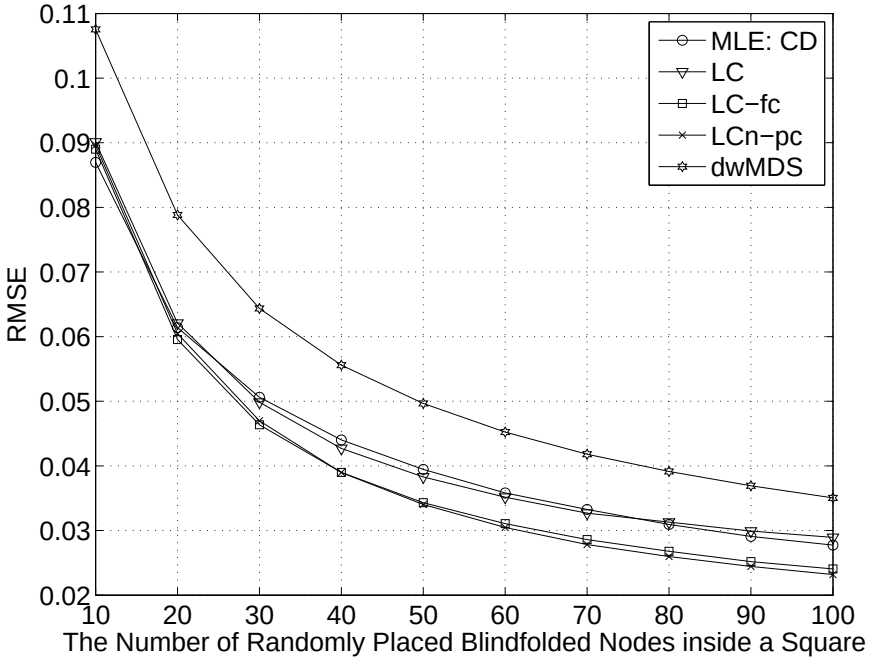

Figure 3.11: RMSE of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 5$ and $\sigma = 5$.

Without loss of generality, p_0 and d_0 are assumed to be 1. The d_{th} can be viewed as a reachable distance of the network without shadowing. The larger the d_{th}, the more connected the nodes are.

First all, Figure 3.16 illustrates how the connected nodes increase as power increases. Different channel parameters also affect the number of connected nodes as shown in Figure 3.17. It is found the low SNR ($n_p = 1.5$) has more connectivity in lower power and less connectivity in higher power than high SNR. Moreover, the plot of average connected anchors is independent of the number of blindfolded nodes because they are placed according to a uniform distribution.

The simulation results, given 20 blindfolded nodes with $n_p = 3$ and $\sigma = 5$, are in Figures 3.18 and 3.19. As d_{th} increases, the estimates of LCs and MLE:CD become

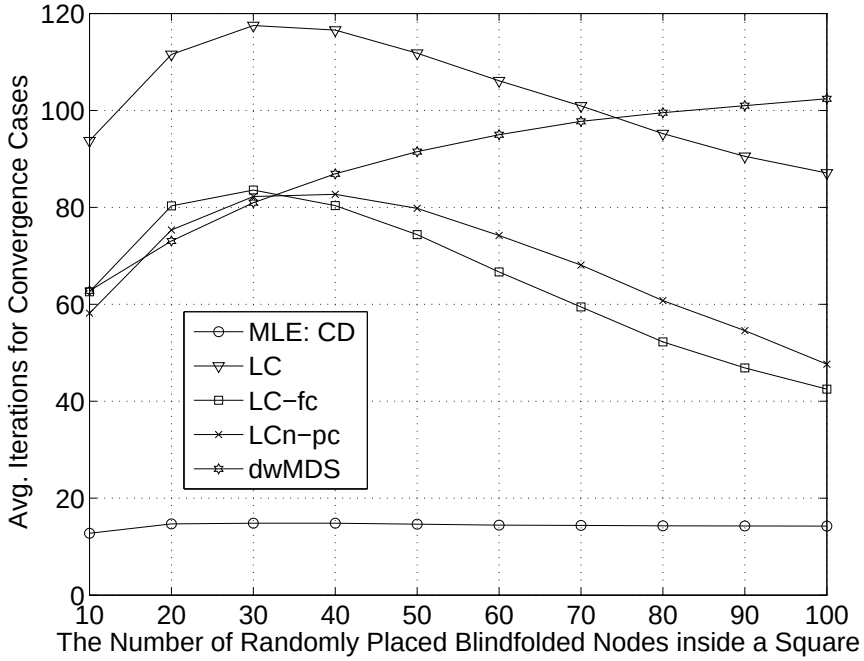

Figure 3.12: Average iterations of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 5$ and $\sigma = 5$.

more accurate. However, this is not true for the dwMDS since $d_{\text{th}} = 1.7$ has the lowest RMSE. More importantly, the dwMDS performs great when the connections are few. The reason is that both LC and MLE require a transmitter "physically" connected to help receivers locate themselves but the dwMDS does not. In dwMDS, the node can guess where other nodes are by its and others' estimated locations. On the other hand, LC family and MLE require range estimates to help others. LC estimators can be modified to have a "virtual" range estimates in the future. The number of iterations decrease as the d_{th} increases. The MLE:CD still has the fewest number of iterations and LC has the most in any connectivity. LCn-pc requires slightly more iterations than dwMDS. The MLE:CD, LC, and dwMDS convergence under all connectivity and LCn-pc has 1 or 2 non-convergent result in $d_{\text{th}} = 1.5$ to

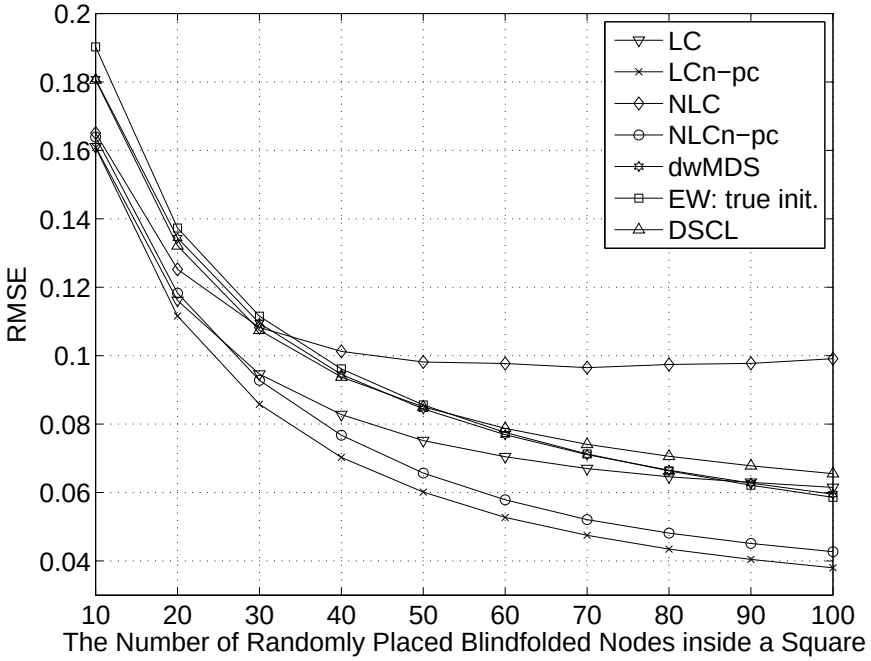

Figure 3.13: RMSE of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

1.7.

The case of 60 blindfolded nodes are shown in Figures 3.20 and 3.21. The dwMDS still works well in low connectivity but the gap is not significant. The number of iterations are similar to the 20 blindfolded nodes.

The low SNR case by given $n_p = 1.5$ and 20 blindfolded nodes is presented in Figure 3.22. The RMSE of dwMDS is again lower than all other methods. Thus dwMDS is suitable for low SNR and low connectivity situations where LCs have a disadvantage. Figure 3.23 shows that dwMDS still works well in low connectivity in the high SNR ($n_p = 5$) and 20 blindfolded nodes. However, as the connectivity increases in either increase transmitting power or adding more nodes, the accuracy of dwMDS is far better than other methods as shown in Figure 3.24.

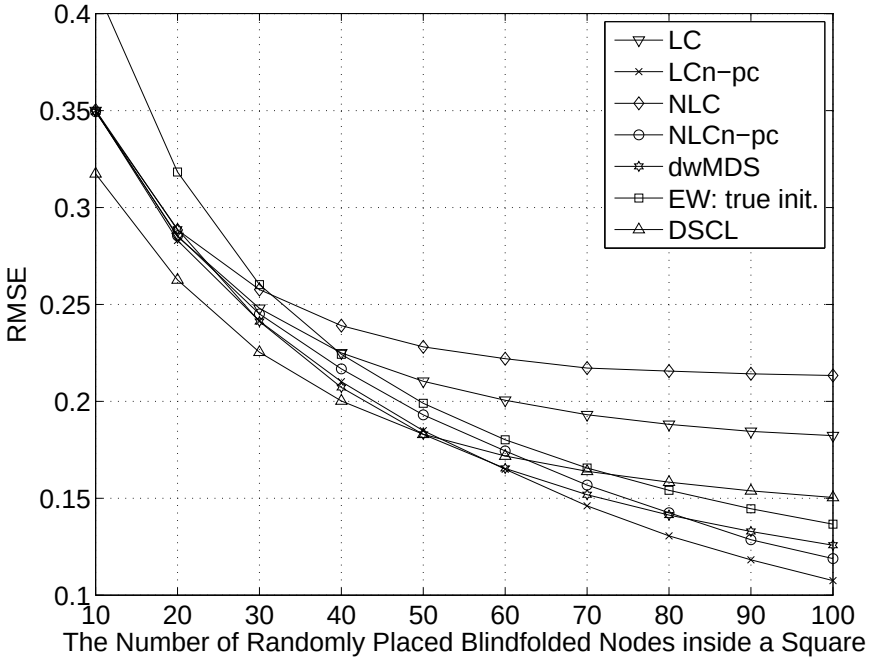

Figure 3.14: RMSE of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 1.5$ and $\sigma = 5$.

3.4.3 Randomly Placed Nodes inside a Unit Circle

In two previous subsections, the positions of anchors are fixed at the corners of a unit square and the number of anchors are 4. In this subsection, blindfolded nodes are still randomly placed with a uniform distribution, but within a unit square. The only exception is the node 1 at the origin for further analysis of connectivity. The anchors now are also randomly placed as blindfolded nodes and different numbers of anchors are investigated. The full connectivity is assumed and the partial connectivity can be extended if needed.

The same three localization methods: the MLE: CD, LC (LC and LCn-pc), and dwMDS, are investigated under 10^4 trails. The initial estimators of three algorithms are the same as the ones in unit square examples with a small change of setting.

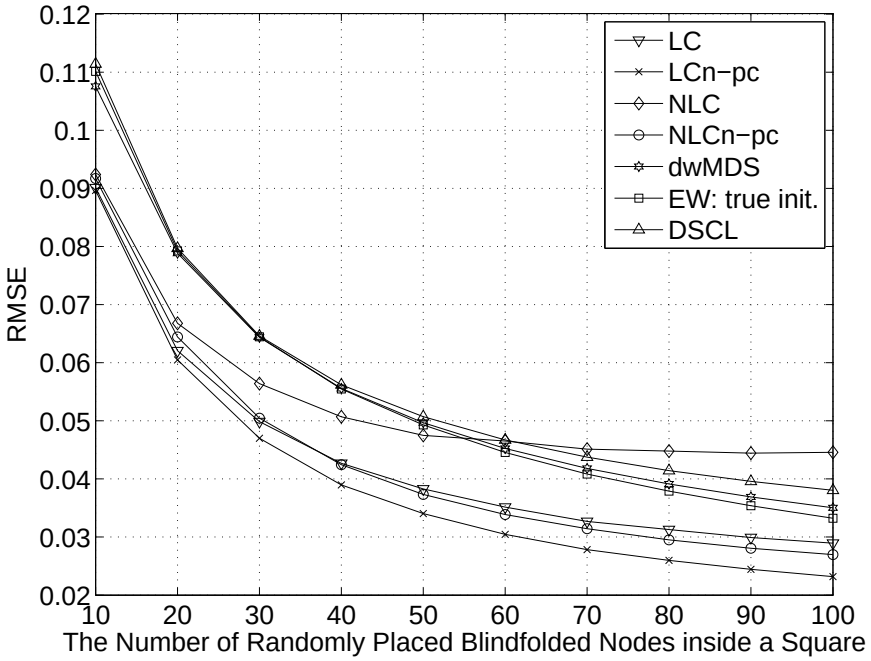

Figure 3.15: RMSE of various localization methods versus the number of randomly placed blindfolded nodes inside a unit square with $n_p = 5$ and $\sigma = 5$.

The setup size of MLE:CD is still 0.005 but the search spaces are $\mathcal{S}_{x_i} = \mathcal{S}_{y_i} = \{-1, -0.995, \ldots, 0.995, 1\}$ which has 400 candidates in each coordinate. The initial CD uses the stopping $\epsilon = 10^{-6}$ in (2.37) and 100 as the permitted maximum number of iterations for each blindfolded node. The cooperative MLE:CD uses the same stopping criteria as LC in (3.100) and ϵ set to be 10^{-6}, and the permitted maximum number of iterations is 10^4. The same ϵ and the allowed number of iterations are also used in LC and its initial estimator, non-cooperative LC. The partial calibration (pc) for LC applies the same stopping parameters as non-cooperative LC. Again, LCs allow all nodes in the network to have another try at most 10 times by using another random guess from $[-1, 1] \times [-1, 1]$ to compute the initial estimate. In dwMDS, the ϵ of modified STRESS function in (3.112) is 10^{-6} and the iterations of dwMDS is at

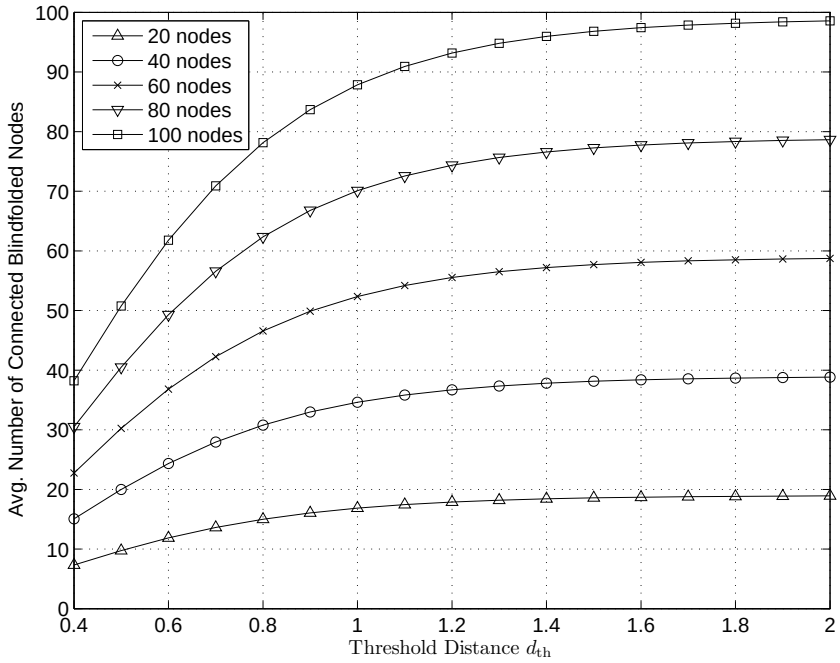

Figure 3.16: Average number of connected blindfolded nodes versus threshold distance inside a unit square with $n_p = 3$ and $\sigma = 5$.

most 10^4.

The simulation results given at most 10 anchors and 40 blindfolded nodes under 10^4 trails are investigated as follows. The LC which converges in all previous examples has a total of 6 unfixed case in 6 different numbers of nodes. The highest percentage is 0.01%. Figure 3.25 shows percentage of non-convergent results for LCn-pc in the unit circle example and the highest percentage is 1.1% when there are 10 anchors and 1 blindfolded node. This one blindfolded node computes the the weight by its estimated location in the cooperative sense which is different from the weight by range estimates in non-cooperative cases. This is the main reason to have unfixed points with low blindfolded nodes. Thus, the convergence is better with fewer anchors and more blindfolded nodes.

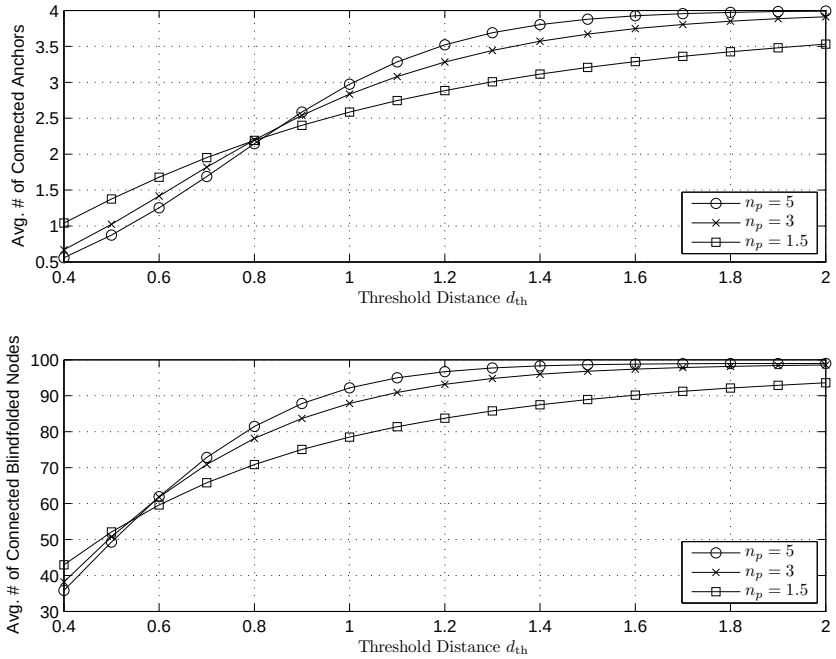

Figure 3.17: Average number of connected anchors and blindfolded nodes given 100 randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

Figures 3.26 and 3.27 illustrate the simulation results for 5 randomly placed anchors and variable numbers of blindfolded nodes. The original LC has the worst RMSE among the 4 presented algorithms. LCn-pc and dwMDS have similar accuracy and LCn-pc outperforms the dwMDS when there are more than 24 blindfolded nodes. The MLE:CD has the lowest RMSE except for 1 blindfolded node where the non-cooperative LC is better as presented in Figure 2.9. The boundaries of exhaustive search in CD helps the accuracy a lot. The number of iterations for LCn-pc is similar to the MLE:CD which is different in the unit square case. One reason is that the ϵ changes from 10^{-8} to 10^{-6}. As the number of anchors goes to 7, LCn-pc starts to have lower RMSE than dwMDS for given any blindfolded nodes as in Figure 3.28 Figures 3.29 and 3.30 show the performance when having 10 randomly

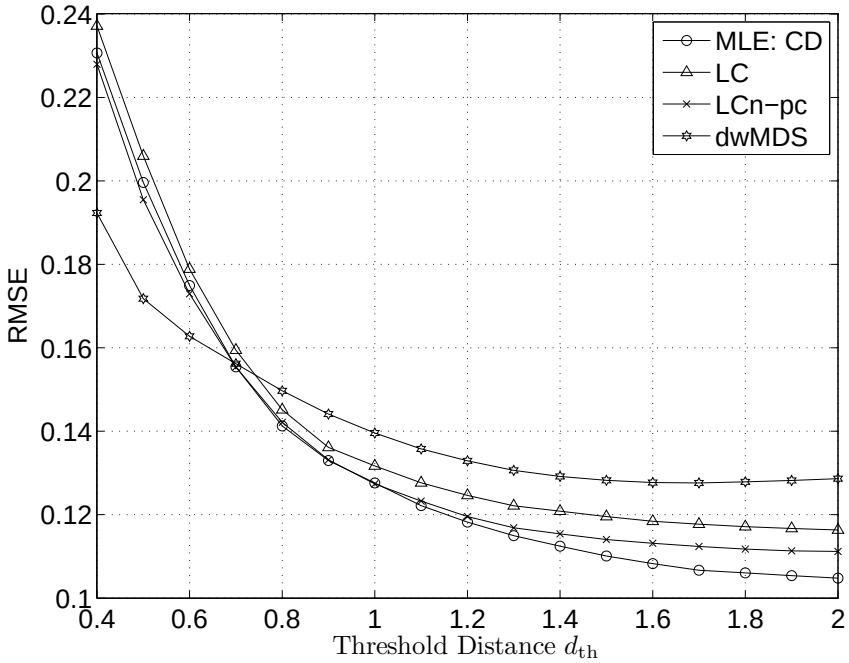

Figure 3.18: RMSE of various methods versus threshold distance given 20 randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

placed anchors. The RMSEs are in the order: LC, dwMDS, LCn-pc, and MLE:CD from highest to lowest. LCn-pc is much closer to MLE:CD in this situation.

3.4.4 Field Test: Motorola Laboratory and DC-10 Aircraft

Two real RSS measurements include the office of Motorola Florida Communications Research Lab, in Plantation, FL [57] and a commercial aircraft DC-10 are presented in this subsection. The measurements from Motorola laboratory has been viewed as a benchmark for many localization research. There are 44 sensors in the network and the four sensors near the edge are treated as anchors. The resulting RMSE for several algorithms are presented in Table 3.1. The MLE here is centralized, not distributed such as coordinate decent, that has the best accuracy. The LCn-pc has the best RMSE among the discussed distributed algorithms and the accuracy of

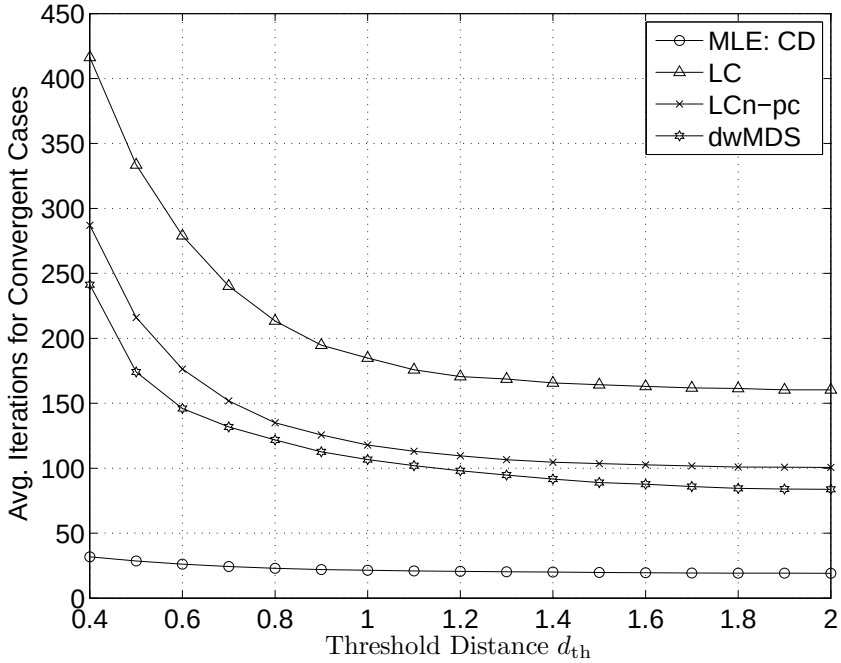

Figure 3.19: Average iterations of various methods versus threshold distance given 20 randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

Table 3.1: RMSE (meter) of localization methods in Motorola laboratory

Method	MLE	dwMDS	LC	LCn-pc	DSCL
RMSE	2.18	2.48	2.38	2.29	2.31

dwMDS is the worst.

Locating three-dimensional positions inside the cabinets of an aircraft DC-10 by the linear combination location estimator is also presented in this subsection. The ITU document "Intra-Aircraft Radio Propagation Analysis And Channel Gain Modeling" [1] shows a measurement campaign regarding the propagation of radio waves in a typical commercial passenger aircraft. This document originally assists

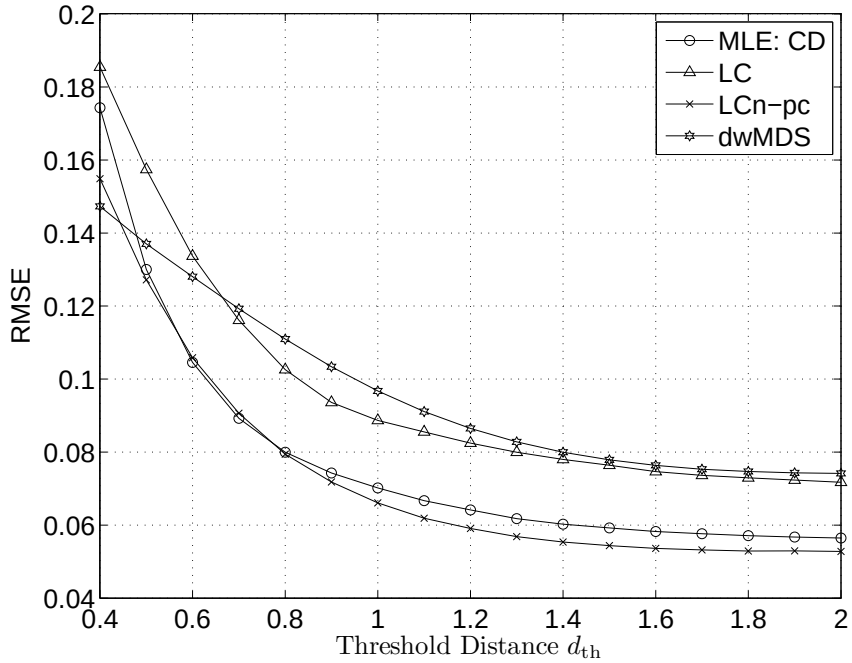

Figure 3.20: RMSE of various localization methods versus threshold distance given 60 randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

to design wireless avionics intra communications (WAIC) and the measured signal strength can also be used to understand localization in an aircraft. The details of test set-up and obtained measurements can be referred to in the same document.

The large scale channel to receive signal strength is now modeled as

$$P_{\text{dB}} = C_0 - 10n_p \log_{10}(d) - 10k_p \log_{10}(f) + Z \tag{3.114}$$

where C_0 is a constant, n_p and k_p are the propagation loss exponents for distance and frequency, respectively, d is the distance in meter, f is the transmitting frequency in MHz, and Z is a zero-mean Gaussian random variable with variance σ^2. The parameters C_0, n_p, and k_p are jointly optimized by the three-dimensional least square method with test point locations, frequency samples, and the received signal strength

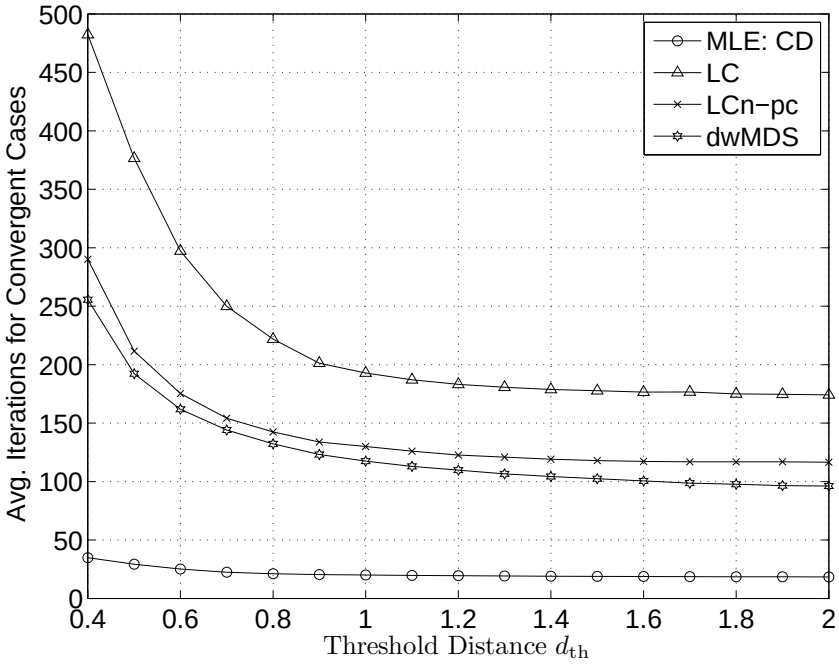

Figure 3.21: Average iterations of various methods versus threshold distance given 60 randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

Table 3.2: Channel Model Parameters for Aircraft DC-10

Group	C_0	k_p	n_p	σ
A	185.5998	2.4014	2.0071	6.6822
B	167.4029	2.0847	3.5012	7.4023

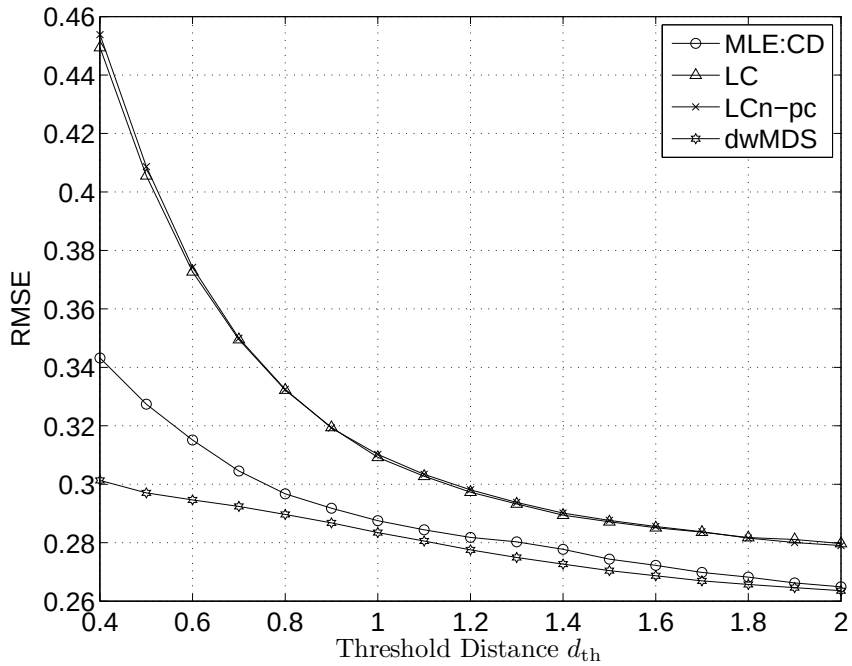

Figure 3.22: RMSE of various localization methods versus threshold distance given 20 randomly placed blindfolded nodes inside a unit square with $n_p = 1.5$ and $\sigma = 5$.

corresponding to distance and frequency to minimize the sum of the squared errors between the received power and predicted large scale power [58]. The variance σ is calculated by the maximum likelihood estimator similar to (2.107):

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n} \left(p_{i,\text{dB}} - \hat{C}_0 + 10\hat{n}_p \log_{10} d_i + 10\hat{k}_p \log_{10} f_i \right)^2}{n}}. \tag{3.115}$$

where n is the number of power measurement. The resulting parameters are given in Table 3.2 where groups A and B represent nodes with the intra-compartment connection and inter-compartment connection, respectively.

The 16 illustrated locations are in PCBUS, PCECN1, and PCECN2 in Figure 3.31. The dash lines represent the connections between test nodes and the PC-

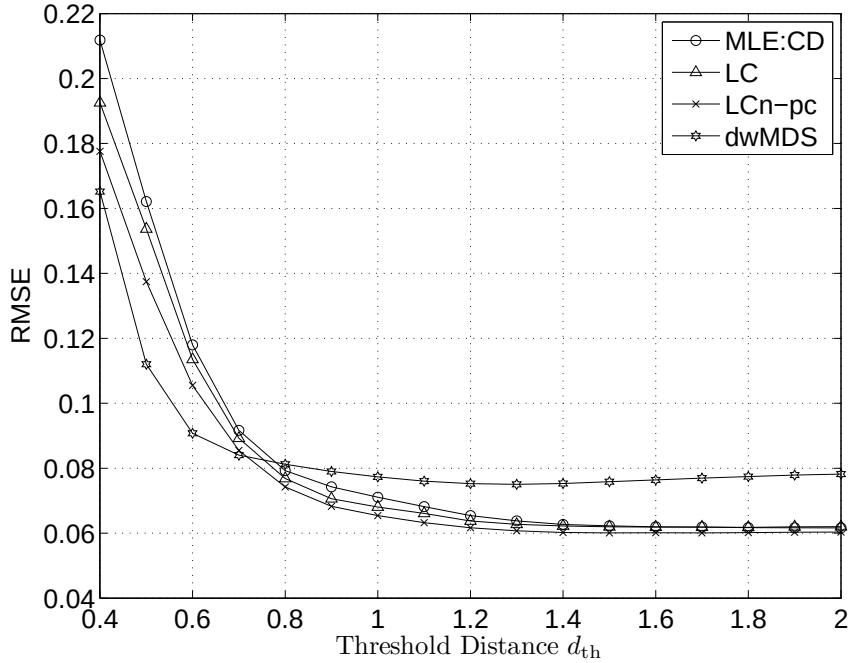

Figure 3.23: RMSE of various localization methods versus threshold distance given 20 randomly placed blindfolded nodes inside a unit square with $n_p = 5$ and $\sigma = 5$.

BUSTP4 connects to all other 15 nodes. The first example considers the square node to be anchors: PCBUSTP4, PCBUSTP7, PCECN1TP1, PCECN1TP7, PCECN2TP6, and PCECN2TP7. Thus all the other 9 nodes are blindfolded nodes where each blindfolded node connects to two anchors and one blindfolded node. The second example assigns 4 more nodes denoted with triangle, PCBUSTP5, PCECN1TP3, PCECN1TP4, PCECN2TP1, and PCECN2PT2 to be anchors. Then each of the rest 5 blindfolded nodes connects 3 anchors. Because frequency is sampled by 20,033 samples within 962 MHz to 18 GHz, the RMSE of each blindfolded node is averaged over 20,333 frequency samples. Moreover, $p_{0,\text{dB}} = C_0 - 10k_p \log_{10} f$ with $d_0 = 1$ meter. The first example applies the bounded cooperative LC estimator and the the second one uses the bounded non-cooperative LC estimator. The boundaries are set by using

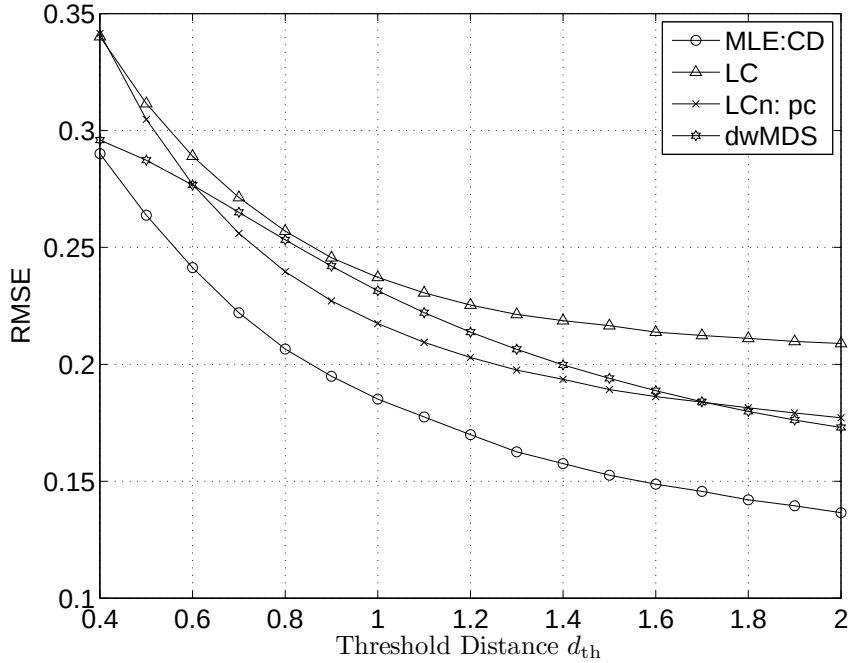

Figure 3.24: RMSE of various localization methods versus threshold distance given 60 randomly placed blindfolded nodes inside a unit square with $n_p = 1.5$ and $\sigma = 5$.

the minimal and maximum coordinates of 16 test nodes to provide better estimates. The volumes with these 6 vertices is given by $33.05 \times 4.06 \times 1.37 = 184.2$ meter^3. Then the RMSE are shown in Table 3.3. The average RMSE for 9 blindfolded nodes in the 2-anchor example is 4.4823 meter and the one for 5 blindfolded nodes in the 3-anchor example is 3.1386 meter.

Because of the lack of real measurements results very mere connectivity in the aircraft example, simulated measurements are provided to investigate the LC algorithms. The 16 nodes still serve as the observed nodes and they are fully connected. The average RMSE versus number of anchors for different LC methods is shown in Figure 3.32. The prefix "B" denotes a bounded algorithm which uses the same boundary as the previous example. The blindfold nodes becomes anchors in the order: PC-

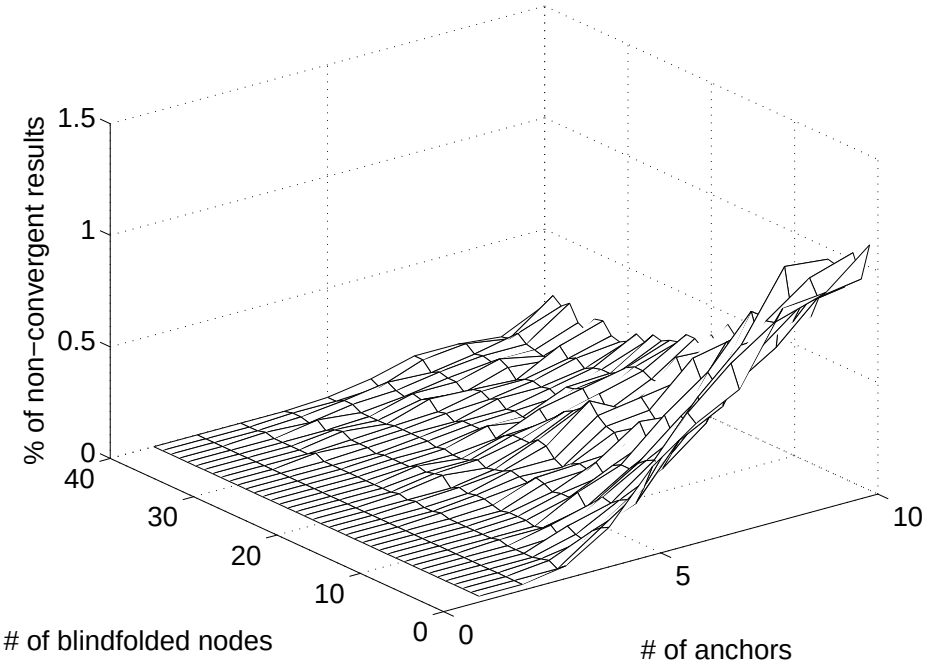

Figure 3.25: Non-convergence percentages of LCn-pc versus the number of randomly placed anchors and blindfolded nodes inside a unit circle with $n_p = 3$ and $\sigma = 5$.

Table 3.3: RMSE (meter) of locating test points inside Aircraft DC-10

	PCBUSTP6	PCECN1TP2	PCECN1TP5	PCECN2TP3	PCECN2TP4
2 anchors	2.9127	2.2426	4.0092	6.6279	4.9514
3 anchors	2.5543	1.4315	2.6889	4.9088	4.1094

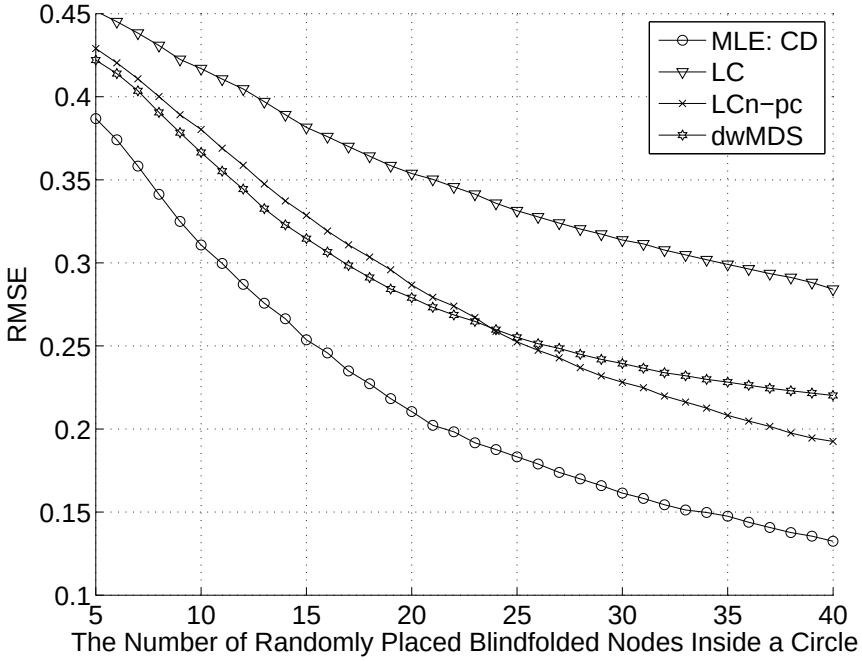

Figure 3.26: RMSE of various localization methods versus the number of randomly placed blindfolded nodes and 5 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

BUSTP8, PCECN1TP2, PCBUSTP6, PCECN2TP3, PCECN1TP5, PCECN2TP4, PCECN1TP8, PCBUSTP4, PCBUSTP7, PCECN2TP9, PCECN1TP3, PCECN1TP4, PCECN2TP6, PCECN2TP2, PCECN2TP5, PCECN1TP1, PCECN1TP6, PCECN2TP1, PCECN1TP7, PCECN2TP8, PCBUSTP5, PCECN2TP7. Clearly, the number of blindfolded nodes decreases as the number of anchors increases. The results shows that the RMSE is less than 3.5 m if there are 5 anchors and less than 2.5 m with 15 anchors. Another important result indicates that the cooperation may hurt the accuracy if the estimated locations of blindfolded nodes is bad.

3.5 Analysis of Estimation Error

This section provides an upper bound and an approximation for cooperative LC estimator, and compares them with the lower bound while developing the algorithm

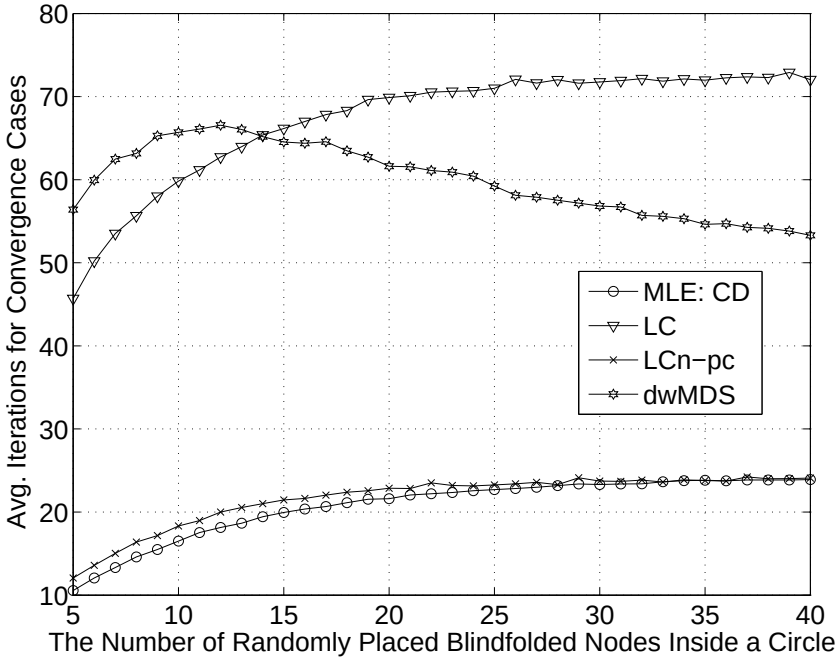

Figure 3.27: Average iterations of various methods versus the number of blindfolded nodes and 5 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

with true directions as shown in Section 3.3.1. Both upper bound and the approximation are derived by randomly guessed directions. Moreover, the equal combining weights are applied in the upper bound; the approximation uses the modified effective distances containing true distances and the higher penalties from the previous iteration of approximation. Although the simulations will show that the approximation can be viewed as an upper bound in most cases, it is inappropriate to name it an upper bound because it contains true distances. The lower bound is correct when all dependence are removed. Thus, the lower bound is slightly lower in most cases and is more accurate in less dense networks. The analysis begins with a fixed topology and then considers a random topology, which can provide simplified results.

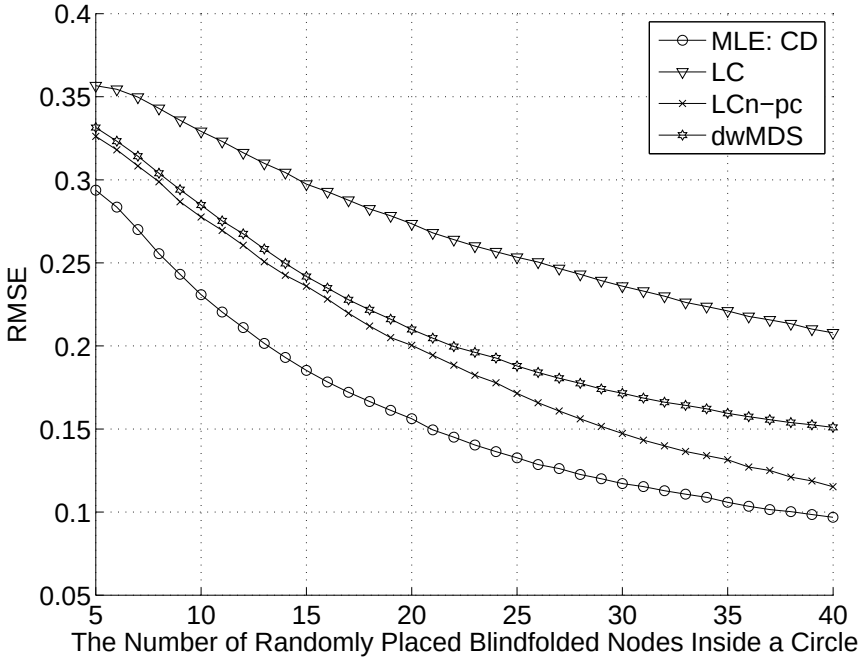

Figure 3.28: RMSE of various localization methods versus the number of randomly placed blindfolded nodes and 7 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

3.5.1 Fixed Topology

The analysis of fixed topology where the true locations are fixed. The two-dimensional case will be considered and applied to other dimensional cases. To simplify the derivation, the network is first assumed to be fully connected and this constraint will be removed. Let $\theta_i = [x_i, y_i]$ for $i = 1, \ldots, m+n$ be the true locations of m blindfold nodes and n anchors. The estimated direction between node i and node j is denoted by $[\cos\hat{\phi}_{ij}, \sin\hat{\phi}_{ij}]$. Then the location estimate for node j from node i in the k-th iteration is given by

$$\hat{\theta}_{ij}^{(k)} = [\hat{x}_{ij}^{(k)}, \hat{y}_{ij}^{(k)}] = [\hat{x}_i^{(k)}, \hat{y}_i^{(k)}] + \hat{d}_{ij}[\cos\hat{\phi}_{ij}, \sin\hat{\phi}_{ij}]. \tag{3.116}$$

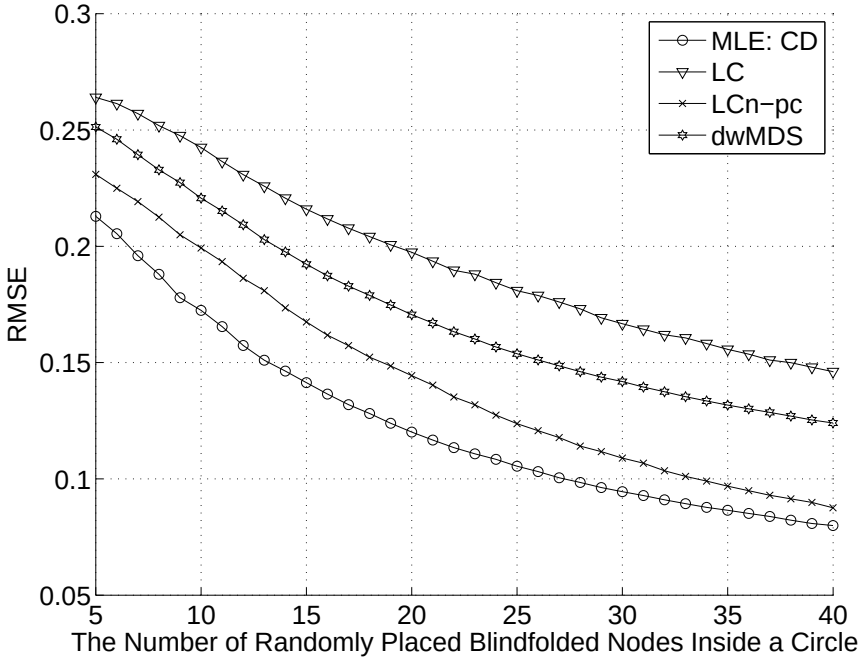

Figure 3.29: RMSE of various localization methods versus the number of randomly placed blindfolded nodes and 10 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

The true location for node j denoted by θ_j can be viewed from node i as

$$\theta_j = [x_j, y_j] = [x_i, y_i] + d_{ij}[\cos\phi_{ij}, \sin\phi_{ij}] \tag{3.117}$$

by replacing estimated location of transmitter, estimated range, and estimated angles with the true ones.

The MSE of blindfolded node j using a linear combination estimator with $\sum_i a_{ij}^{(k)} =$

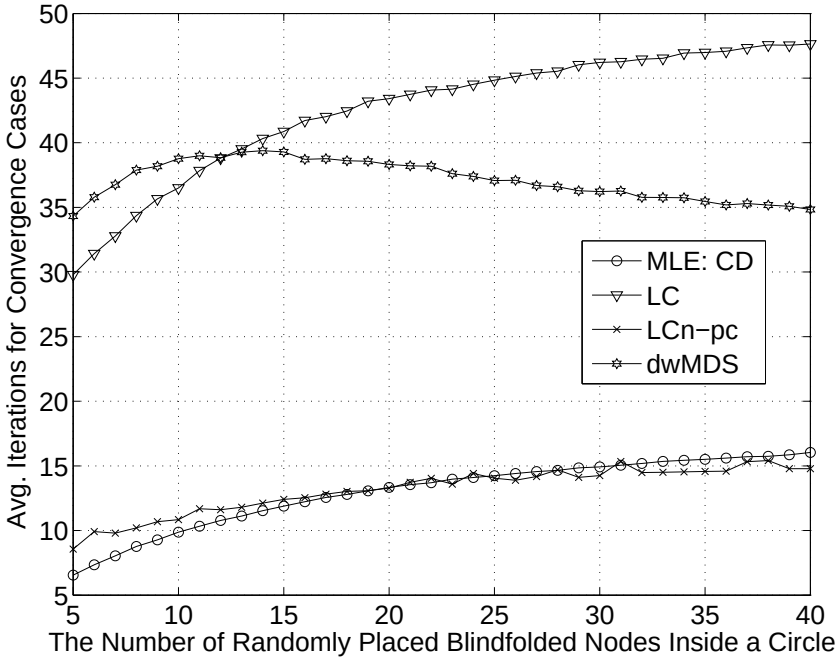

Figure 3.30: Average iterations of various methods versus the number of randomly placed blindfolded nodes and 10 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

1 after k iterations can be written as

$$\begin{aligned}\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) &= \mathbb{E}_{\hat{\theta}_j^{(k+1)}}[\|\hat{\theta}_j^{(k+1)} - \theta_j\|^2] \\ &= \mathbb{E}\left[\left\|\sum_{i=1}^{m+n} a_{ij}^{(k)}\hat{\theta}_{ij}^{(k)} - \theta_j\right\|^2\right] = \mathbb{E}\left[\left\|\sum_{i=1}^{m+n} a_{ij}^{(k)}(\hat{\theta}_{ij}^{(k)} - \theta_j)\right\|^2\right]. \qquad (3.118)\end{aligned}$$

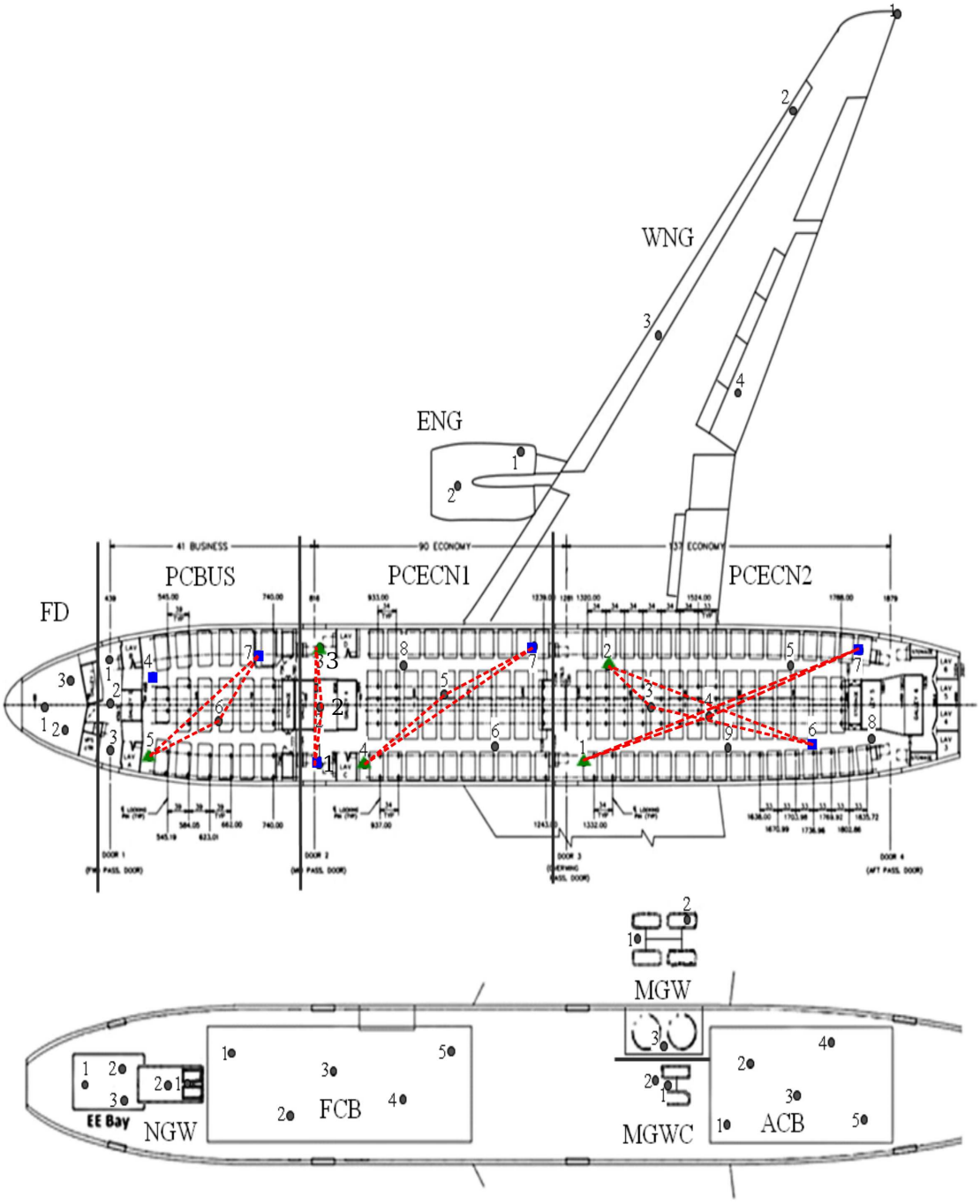

Figure 3.31: Test point locations used in inter- and intra-compartment coupling measurements in DC-10 [1].

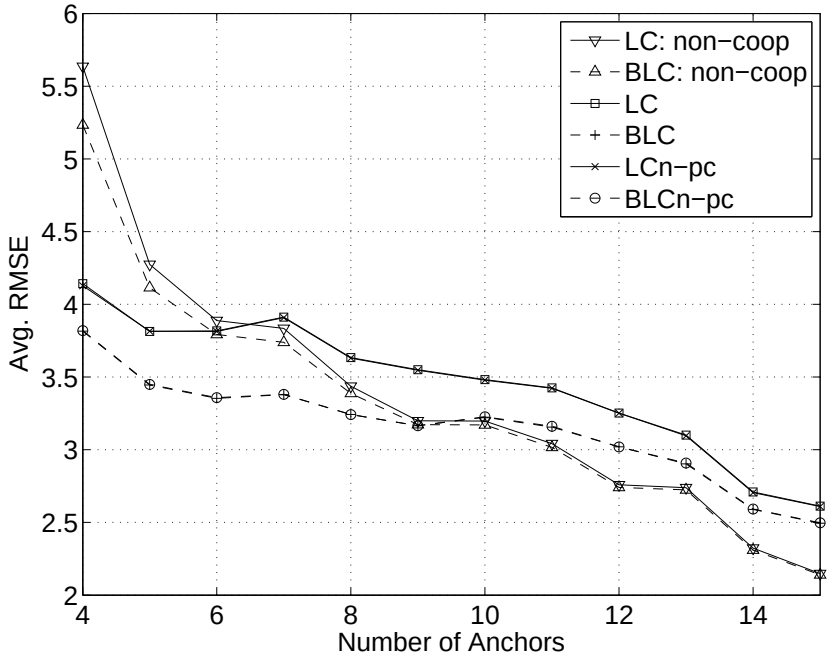

Figure 3.32: Average RMSE of various localization methods versus the number of anchors for a full connectivity in DC-10.

In the 2-D case, the MSE becomes

$$\begin{aligned}
\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) &= \mathbb{E}\left[\left\|\sum_{i=1}^{m+n} a_{ij}^{(k)}[\hat{x}_{ij}^{(k)} - x_j, \hat{y}_{ij}^{(k)} - y_j]\right\|^2\right] \\
&= \mathbb{E}\left[\left(\sum_{i=1}^{m+n} a_{ij}^{(k)}(\hat{x}_{ij}^{(k)} - x_j)\right)^2 + \left(\sum_{i=1}^{m+n} a_{ij}^{(k)}(\hat{y}_{ij}^{(k)} - y_j)\right)^2\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{m+n}\sum_{l=1}^{m+n} a_{ij}^{(k)}[(\hat{x}_{ij}^{(k)} - x_j)(\hat{x}_{lj}^{(k)} - x_j) + (\hat{y}_{ij}^{(k)} - y_j)(\hat{y}_{lj}^{(k)} - y_j)]a_{lj}^{(k)}\right] \\
&\equiv \mathbb{E}[(\mathbf{a}^{(k)})^{\mathrm{T}}\mathbf{M}^{(k)}\mathbf{a}^{(k)}] \qquad (3.119)
\end{aligned}$$

where $\mathbf{a}^{(k)} = \left[a_{1j}^{(k)} \cdots a_{m+n,j}^{(k)}\right]^{\mathrm{T}}$ and $\mathbf{M}^{(k)}$ is an $m+n$ by $m+n$ matrix. If the weights are fixed for a given topology, i.e., the weights are independent of location estimates,

$\mathbf{a}^{(k)}$ is no longer a random vector and thus

$$\begin{aligned}\mathrm{MSE}(\hat{\theta}_j) &= \mathbb{E}[(\mathbf{a}^{(k)})^{\mathrm{T}}\mathbf{M}^{(k)}\mathbf{a}^{(k)}] = (\mathbf{a}^{(k)})^{\mathrm{T}}\mathbb{E}[\mathbf{M}^{(k)}]\mathbf{a}^{(k)} \\ &= \sum_{i=1}^{m+n}(a_{ij}^{(k)})^2\mathbb{E}[\mathbf{M}_{ii}^{(k)}] + \sum_{i=1}^{m+n}\sum_{l=1,l\neq i}^{m+n} a_{ij}^{(k)}a_{lj}^{(k)}\mathbb{E}[\mathbf{M}_{il}^{(k)}] \end{aligned} \tag{3.120}$$

For example, the weights in the proposed linear combination estimator:

$$a_{ij}^{(k)} = \frac{1/(\delta_{ij}^{(k)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(k)})^2} \tag{3.121}$$

where $\delta_{ij}^{(k)} = \sqrt{\gamma_{ij}^{(k-1)} + d_{ij}^2/h_{ij}}$ and the penalty $\gamma_{ij}^{(k-1)}$ depends on the estimates at the $(k-1)$-th stage, not the k-th stage. Another example is the equal weight, i.e., $a_{ij}^{(k)} = 1/(m+n)$ where $m+n$ is the total number of connected node to j. The corresponding upper bounds will be derived according to the two above examples.

The elements of matrix $\mathbf{M}^{(k)}$: $\mathbf{M}_{il}^{(k)}$ $(i \neq l)$ and $\mathbf{M}_{ii}^{(k)}$ can be written as

$$\begin{aligned}
\mathbf{M}_{il}^{(k)} &= (\hat{x}_i^{(k)} + \hat{d}_{ij}\cos\hat{\phi}_{ij} - x_i - d_{ij}\cos\phi_{ij})(\hat{x}_l^{(k)} + \hat{d}_{lj}\cos\hat{\phi}_{lj} - x_l - d_{lj}\cos\phi_{lj}) \\
&\quad + (\hat{y}_i^{(k)} + \hat{d}_{ij}\sin\hat{\phi}_{ij} - y_i - d_{ij}\sin\phi_{ij})(\hat{y}_l^{(k)} + \hat{d}_{lj}\sin\hat{\phi}_{lj} - y_l - d_{lj}\sin\phi_{lj}) \\
&= \hat{x}_i^{(k)}\hat{x}_l^{(k)} + \hat{x}_i^{(k)}\hat{d}_{lj}\cos\hat{\phi}_{lj} - \hat{x}_i^{(k)}x_l - \hat{x}_i^{(k)}d_{lj}\cos\phi_{lj} + \hat{x}_l^{(k)}\hat{d}_{ij}\cos\hat{\phi}_{ij} \\
&\quad + \hat{d}_{ij}\hat{d}_{lj}\cos\hat{\phi}_{ij}\cos\hat{\phi}_{lj} - x_l\hat{d}_{ij}\cos\hat{\phi}_{ij} - \hat{d}_{ij}d_{lj}\cos\hat{\phi}_{ij}\cos\phi_{lj} \\
&\quad - x_i\hat{x}_l^{(k)} - x_i\hat{d}_{lj}\cos\hat{\phi}_{lj} + x_ix_l + x_id_{lj}\cos\phi_{lj} - \hat{x}_l^{(k)}d_{ij}\cos\phi_{ij} \\
&\quad - d_{ij}\hat{d}_{lj}\cos\phi_{ij}\cos\hat{\phi}_{lj} + x_ld_{ij}\cos\phi_{ij} + d_{ij}d_{lj}\cos\phi_{ij}\cos\phi_{lj} \\
&\quad + \hat{y}_i^{(k)}\hat{y}_l^{(k)} + \hat{y}_i^{(k)}\hat{d}_{lj}\sin\hat{\phi}_{lj} - \hat{y}_i^{(k)}y_l - \hat{y}_i^{(k)}d_{lj}\sin\phi_{lj} + \hat{y}_l^{(k)}\hat{d}_{ij}\sin\hat{\phi}_{ij} \\
&\quad + \hat{d}_{ij}\hat{d}_{lj}\sin\hat{\phi}_{ij}\sin\hat{\phi}_{lj} - y_l\hat{d}_{ij}\sin\hat{\phi}_{ij} - \hat{d}_{ij}d_{lj}\sin\hat{\phi}_{ij}\sin\phi_{lj} \\
&\quad - y_i\hat{y}_l^{(k)} - y_i\hat{d}_{lj}\sin\hat{\phi}_{lj} + y_iy_l + y_id_{lj}\sin\phi_{lj} - \hat{y}_l^{(k)}d_{ij}\sin\phi_{ij} \\
&\quad - d_{ij}\hat{d}_{lj}\sin\phi_{ij}\sin\hat{\phi}_{lj} + y_ld_{ij}\sin\phi_{ij} + d_{ij}d_{lj}\sin\phi_{ij}\sin\phi_{lj}, \qquad (3.122)
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{M}_{ii}^{(k)} &= (\hat{x}_i^{(k)} + \hat{d}_{ij}\cos\hat{\phi}_{ij} - x_i - d_{ij}\cos\phi_{ij})^2 + (\hat{y}_i^{(k)} + \hat{d}_{ij}\sin\hat{\phi}_{ij} - y_i - d_{ij}\sin\phi_{ij})^2 \\
&= (\hat{x}_i^{(k)} - x_i)^2 + (\hat{d}_{ij}\cos\hat{\phi}_{ij} - d_{ij}\cos\phi_{ij})^2 + 2(\hat{x}_i^{(k)} - x_i)(\hat{d}_{ij}\cos\hat{\phi}_{ij} - d_{ij}\cos\phi_{ij}) \\
&\quad + (\hat{y}_i^{(k)} - y_i)^2 + (\hat{d}_{ij}\sin\hat{\phi}_{ij} - d_{ij}\sin\phi_{ij})^2 + 2(\hat{y}_i^{(k)} - y_i)(\hat{d}_{ij}\sin\hat{\phi}_{ij} - d_{ij}\sin\phi_{ij}) \\
&= (\hat{x}_i^{(k)} - x_i)^2 + (\hat{y}_i^{(k)} - y_i)^2 + \hat{d}_{ij}^2 + d_{ij}^2 - 2\hat{d}_{ij}d_{ij}(\cos\hat{\phi}_{ij}\cos\phi_{ij} + \sin\hat{\phi}_{ij}\sin\phi_{ij}) \\
&\quad + 2(\hat{x}_i^{(k)} - x_i)(\hat{d}_{ij}\cos\hat{\phi}_{ij} - d_{ij}\cos\phi_{ij}) + 2(\hat{y}_i^{(k)} - y_i)(\hat{d}_{ij}\sin\hat{\phi}_{ij} - d_{ij}\sin\phi_{ij}).
\end{aligned} \qquad (3.123)$$

The plain case of angle estimation $\hat{\phi}_{ij}$ is a uniform guess in $[0, 2\pi)$, which means the estimator has no other information. If any methods of angle estimation with a larger error than the random guess under certain environment, it should be replaced by

the random guess to ensure the worse case valid. The random guess makes $\hat{\phi}_{ij}$ and $\hat{\phi}_{lj}$ independent of all other estimated variables. Then $\mathbb{E}[\mathbf{M}_{ii}]$ and $\mathbb{E}[\mathbf{M}_{il}]$ contains $\hat{\phi}_{ij}$ and $\hat{\phi}_{lj}$ can be products of expectations. Because $\hat{\phi}_{ij}$ and $\hat{\phi}_{lj}$ are uniformly distributed in an interval $[0, 2\pi)$, $\mathbb{E}[\cos\hat{\phi}_{ij}] = 0$ and $\mathbb{E}[\sin\hat{\phi}_{ij}] = 0$, which make the expectations containing $\hat{\phi}_{ij}$ or $\hat{\phi}_{lj}$ also 0. Therefore,

$$\begin{aligned}\mathbb{E}[\mathbf{M}_{il}^{(k)}] &= \mathbb{E}[(\hat{x}_i^{(k)} - x_i)(\hat{x}_l^{(k)} - x_l)] + \mathbb{E}[(\hat{y}_i^{(k)} - y_i)(\hat{y}_l^{(k)} - y_l)] \\ &\quad + (x_i - \mathbb{E}[\hat{x}_i^{(k)}])d_{lj}\cos\phi_{lj} + (x_l - \mathbb{E}[\hat{x}_l^{(k)}])d_{ij}\cos\phi_{ij} \\ &\quad + (y_i - \mathbb{E}[\hat{y}_i^{(k)}])d_{lj}\sin\phi_{lj} + (y_l - \mathbb{E}[\hat{y}_l^{(k)}])d_{ij}\sin\phi_{ij} \\ &\quad + d_{ij}d_{lj}(\cos(\phi_{ij})\cos(\phi_{lj}) + \sin(\phi_{ij})\sin(\phi_{lj})) \qquad (3.124) \\ \mathbb{E}[\mathbf{M}_{ii}^{(k)}] &= \mathrm{MSE}(\hat{x}_i^{(k)}) + \mathrm{MSE}(\hat{y}_i^{(k)}) + \mathbb{E}[\hat{d}_{ij}^2] + d_{ij}^2 \\ &\quad + 2(x_i - \mathbb{E}[\hat{x}_i^{(k)}])d_{ij}\cos\phi_{ij} + 2(y_i - \mathbb{E}[\hat{y}_i^{(k)}])d_{ij}\sin\phi_{ij} \qquad (3.125)\end{aligned}$$

To simplify the equation and prepare to remove full connectivity, let

$$\mathbf{K}_{il}^{(k)} \equiv \mathbb{E}[(\hat{x}_i^{(k)} - x_i)(\hat{x}_l^{(k)} - x_l)]/h_{ij}/h_{lj} + \mathbb{E}[(\hat{y}_i^{(k)} - y_i)(\hat{y}_l^{(k)} - y_l)]/h_{ij}/h_{lj} \qquad (3.126)$$

and

$$\mathbf{R}_{ij}^{(k)} \equiv 2d_{ij}/h_{ij}\left[(x_i - \mathbb{E}[\hat{x}_i^{(k)}])\cos\phi_{ij} + (y_i - \mathbb{E}[\hat{y}_i^{(k)}])\sin\phi_{ij}\right]. \qquad (3.127)$$

Applying the statistics of unbiased range estimates: $\mathbb{E}[\hat{d}_{ij}^2] = d_{ij}^2\exp(b^2)$ and removing

the assumption of full connection, the $\mathbb{E}[\mathbf{M}^{(k)}]$ becomes

$$\begin{aligned}\mathbb{E}[\mathbf{M}_{il}^{(k)}] &= \mathbf{K}_{il}^{(k)} + d_{ij}/h_{ij}d_{lj}/h_{lj}(\cos(\phi_{ij})\cos(\phi_{lj}) + \sin(\phi_{ij})\sin(\phi_{lj})) \\ &\quad + (x_i - \mathbb{E}[\hat{x}_i^{(k)}])d_{lj}/h_{lj}\cos\phi_{lj} + (x_l - \mathbb{E}[\hat{x}_l^{(k)}])d_{ij}/h_{ij}\cos\phi_{ij} \\ &\quad + (y_i - \mathbb{E}[\hat{y}_i^{(k)}])d_{lj}/h_{lj}\sin\phi_{lj} + (y_l - \mathbb{E}[\hat{y}_l^{(k)}])d_{ij}/h_{ij}\sin\phi_{ij} \qquad (3.128) \\ \mathbb{E}[\mathbf{M}_{ii}^{(k)}] &= \mathrm{MSE}(\hat{\theta}_i^{(k)}) + \left[\exp(b^2)+1\right] d_{ij}^2/h_{ij} + \mathbf{R}_{ij}^{(k)} \\ &\equiv [\exp(b^2)+1](\delta_{ij}^{(k)})^2 + \mathbf{R}_{ij}^{(k)}. \qquad (3.129)\end{aligned}$$

The effective distance here are modified as

$$\delta_{ij}^{(k)} = \sqrt{\frac{\mathrm{MSE}(\hat{\theta}_i^{(k)})}{\exp(b^2)+1} + \frac{d_{ij}^2}{h_{ij}}}, \qquad (3.130)$$

which is different from the one in lower bound related to the lowest variance. Then the (generic) upper bound for cooperative LC at k-stage can be expressed as

$$\begin{aligned}\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) &\le \sum_{i=1}^{m+n}(a_{ij}^{(k)})^2\mathbb{E}[\mathbf{M}_{ii}^{(k)}] + \sum_{i=1}^{m+n}\sum_{l=1,l\neq i}^{m+n} a_{ij}^{(k)}a_{lj}^{(k)}\mathbb{E}[\mathbf{M}_{il}^{(k)}] \\ &= [\exp(b^2)+1]\sum_{i=1}^{m+n}(a_{ij}^{(k)}\delta_{ij}^{(k)})^2 + \sum_{i=1}^{m+n}(a_{ij}^{(k)})^2\mathbf{R}_{ij}^{(k)} \\ &\quad + \sum_{i=1}^{m+n}\sum_{l=1,l\neq i}^{m+n} a_{ij}^{(k)}a_{lj}^{(k)}\mathbb{E}[\mathbf{M}_{il}^{(k)}] \qquad (3.131)\end{aligned}$$

If the combining weights are assigned by the (modified) effective distances:

$$a_{ij}^{(k)} = \frac{1/(\delta_{ij}^{(k)})^2}{\sum_{l=1}^{m+n} 1/(\delta_{lj}^{(k)})^2}, \qquad (3.132)$$

the approximation of the cooperative LC estimator is provided as

$$\begin{aligned}\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) &\leq \sum_{i=1}^{m+n} \frac{\left(\frac{1}{(\delta_{ij}^{(k)})^2}\right)^2 (\delta_{ij}^{(k)})^2 \left[\exp(b^2)+1\right]}{(\sum_{q=1}^{m+n} 1/(\delta_{qj}^{(k)})^2)^2} + \sum_{i=1}^{m+n} (a_{ij}^{(k)})^2 \mathbf{R}_{ij}^{(k)} \\ &\quad + \sum_{i=1}^{m+n} \sum_{l=1, l\neq i}^{m+n} a_{ij}^{(k)} a_{lj}^{(k)} \mathbb{E}[\mathbf{M}_{il}^{(k)}] \\ &= \frac{\exp(b^2)+1}{\sum_{q=1}^{m+n} 1/(\delta_{qj}^{(k)})^2} + \sum_{i=1}^{m+n} (a_{ij}^{(k)})^2 \mathbf{R}_{ij}^{(k)} + \sum_{i=1}^{m+n} \sum_{l=1, l\neq i}^{m+n} a_{ij}^{(k)} a_{lj}^{(k)} \mathbb{E}[\mathbf{M}_{il}^{(k)}].\end{aligned} \tag{3.133}$$

It is an approximation because its combining weights are different from the optimal weights in lower bound and in the real LC algorithm. However, substituting the weights by optimal weights or estimated weights slightly changes the upper bound. Another reason is the approximation contains the true distances that are unknown to estimators and should not be used for a worse/upper case. However, this approximation can served as an upper bound of MSE for LC estimators according to simulations. On the other hand, substituting the weight $a_{ij}^{(k)}$ with the equal weight:

$$a_{ij}^{(k)} = \frac{h_{ij}}{\sum_{q=1}^{m+n} h_{qj}} \quad \text{for } \mathrm{MSE}(\hat{\theta}_i^{(k)}) \text{ and } \mathrm{MSE}(\hat{\theta}_q^{(k)}) < \infty. \tag{3.134}$$

in (3.131) results in the upper bound for the cooperative LC estimator. In other words, the upper bound is accomplished by using both the plain cases of combining weights and estimated angles. Thus the upper bound and the approximation can be derived in each stage by using the network topology (θ_i), $\mathrm{MSE}(\hat{\theta}_i^{(k)})$, $\mathbf{K}_{il}^{(k)}$, and

$\mathbb{E}[\hat{\theta}_i^{(k)}]$ where

$$\begin{aligned}\mathbb{E}[\hat{\theta}_i^{(k)}] &= \mathbb{E}\left[\sum_{l=1}^{m+n} a_{li}^{(k-1)}\hat{\theta}_{li}^{(k-1)}\right] = \sum_{l=1}^{m+n} a_{li}^{(k-1)}\mathbb{E}[\hat{\theta}_{li}^{(k-1)}] \\ &= \sum_{l=1}^{m+n} a_{li}^{(k-1)}\left(\mathbb{E}[\hat{\theta}_l^{(k-1)}] + \mathbb{E}[\hat{d}_{li}]\left[\mathbb{E}[\cos\hat{\phi}_{li}], \mathbb{E}[\sin\hat{\phi}_{li}]\right]\right) \\ &= \sum_{l=1}^{m+n} a_{li}^{(k-1)}\mathbb{E}[\hat{\theta}_l^{(k-1)}] \end{aligned} \tag{3.135}$$

is biased in general.

In conclusion, the upper bound and the approximation use iterative methods as well as the lower bound by starting with $\text{MSE}(\hat{\theta}_i^{(k)}) = \infty$ for blindfolded nodes that have not been estimated and $\text{MSE}(\hat{\theta}_i^{(k)}) = 0$ for anchors. Furthermore, both the upper bound and the approximation for $\text{MSE}(\hat{\theta}_i^{(1)})$ are the same as the two corresponding results in non-cooperative LC.

Figure 3.33 shows the lower and upper bounds with nodes at a 6×6 uniform grid of unit square where 4 corner nodes are anchors. To make the computation easy, $\hat{\theta}_i^{(k)}$ and $\hat{\theta}_i^{(k)}$ are assumed to be uncorrelated and therefore

$$\begin{aligned}\mathbf{K}_{il}^{(k)} &\equiv \left[\mathbb{E}[(\hat{x}_i^{(k)} - x_i)(\hat{x}_l^{(k)} - x_l)] + \mathbb{E}[(\hat{y}_i^{(k)} - y_i)(\hat{y}_l^{(k)} - y_l)]\right]/h_{ij}/h_{lj} \\ &= \left[(x_i - \mathbb{E}[\hat{x}_i^{(k)}])(x_l - \mathbb{E}[\hat{x}_l^{(k)}]) + (y_i - \mathbb{E}[\hat{y}_i^{(k)}])(y_l - \mathbb{E}[\hat{y}_l^{(k)}])\right]/h_{ij}/h_{lj}.\end{aligned} \tag{3.136}$$

The above equation is valid for $k = 1$ because $\hat{\theta}_i^{(1)}$ depends on anchors' locations and distances to node i that make $\hat{\theta}_i^{(1)}$ not random. For $k > 1$, the results with and without the uncorrelated assumption are slightly different.

The abbreviation "app." represents the approximation and "upper" is the ordinary upper bound by equal weights. Beside the analytical bounds and approximation derived before, their related simulations are also given and denoted them with the

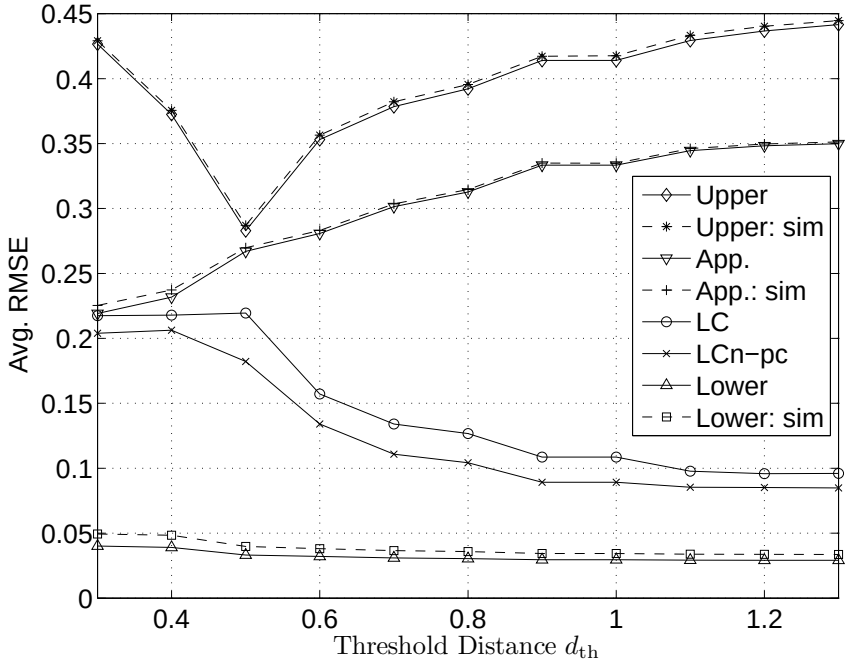

Figure 3.33: RMSE of various localization methods versus threshold distance given anchors and blindfolded nodes in a 6×6 grid of a unit square with $n_p = 3$ and $\sigma = 5$.

abbreviation "sim.". The simulation of the lower bound is executed by giving optimal weight and perfectly knowing directions as well as the lower bound. As expected in the analytical part of the LC estimator, the simulation of the lower bound is slightly higher than the lower bound because the dependence is not completely removed. Similarly, the simulations of the upper bound and approximation apply the weights computed by the corresponding upper bounds and use random directions. Again, the simulated upper results are slightly larger than the analytical upper results because of the dependence. It is also realized that both the upper bound and approximation worsen as the connectivity increases in most cases, because connecting more noisy blindfolded nodes could reduce the accuracy of the estimate.

3.5.2 Random Topology

This subsection investigates the lower bound, the upper bound, and the approximation in random topologies, and the corresponding simplified results will be derived. When the topology is random, i.e., part or all locations of anchors and blindfolded node are also random, and these locations are denoted by Θ_i for $i = 1, \ldots, m+n$. The corresponding distances and directions become random variables and can be denoted by D_{ij} and Φ_{ij}, respectively. The related randoms vectors can be denoted by $\mathbf{\Theta} \equiv [\Theta_1, \ldots, \Theta_{j-1}, \Theta_j, \ldots, \Theta_{m+n}]$, $D \equiv [D_{1j}, \ldots, D_{m+n,j}]$, and $\Phi \equiv [\Phi_{1j}, \ldots, \Phi_{m+n,j}]$. The previous case with fixed locations can be viewed as conditioning on $\Theta_i = \theta_i$. Then the MSE of node j needs to average over $\hat{\theta}_j$ and $\mathbf{\Theta}$. Thus the lower bound is given by knowing the directions:

$$\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) = \mathbb{E}_{\mathbf{\Theta}\hat{\theta}_j^{(k+1)}}[\|\hat{\theta}_j^{(k+1)} - \theta_j\|^2] \geq \mathbb{E}_D \left[\frac{\exp(b^2) - 1}{\sum_{i=1}^{m+n} 1/(\Omega_{ij}^{(k)})^2} \right] \tag{3.137}$$

where

$$(\Omega_{ij}^{(k)})^2 = \frac{1}{\sum_{l=1, l\neq j}^{m+n} \frac{1}{(\Omega_{li}^{(k-1)})^2}} + \frac{D_{ij}^2}{h_{ij}}. \tag{3.138}$$

is the square of (optimal) effective distance and it is a random variable. The result is again irrelevant to the angle vector Φ because it is perfectly known. As shown in the above equations, the expectation of $\mathbf{\Theta}$ can be replaced by D.

Similarly, the generic upper bound is written as

$$\begin{aligned} \mathrm{MSE}(\hat{\theta}_j^{(k+1)}) \leq\ & [\exp(b^2) + 1]\mathbb{E}_{\mathbf{\Theta}} \left[\sum_{i=1}^{m+n} (a_{ij}^{(k)} \Delta_{ij}^{(k)})^2 \right] \\ & + \mathbb{E}_{\mathbf{\Theta}} \left[\sum_{i=1}^{m+n} (a_{ij}^{(k)})^2 \mathbf{R}_{ii}^{(k)} + \sum_{i=1}^{m+n} \sum_{l=1, l\neq i}^{m+n} a_{ij}^{(k)} a_{lj}^{(k)} \mathbb{E}[\mathbf{M}_{il}^{(k)}] \right] \end{aligned} \tag{3.139}$$

where the square of (modified, worse) effective distance

$$(\Delta_{ij}^{(k)})^2 = \frac{\mathrm{MSE}(\hat{\Theta}_i^{(k)})}{\exp(b^2)+1} + \frac{D_{ij}^2}{h_{ij}} \tag{3.140}$$

is also a random variable. If the directions to the target node j are uniformly distributed, e.g., anchors and blindfolded nodes are randomly placed inside a circle, $\mathbb{E}_{\Phi_{ij}}[\cos\Phi_{ij}] = \mathbb{E}_{\Phi_{ij}}[\sin\Phi_{ij}] = 0$. Then

$$\mathbb{E}_{\mathbf{\Theta}}\left[\mathbf{R}_{ii}^{(k)}\right] = 0 \tag{3.141}$$

$$\mathbb{E}_{\mathbf{\Theta}}\left[\mathbb{E}[\mathbf{M}_{il}^{(k)}]\right] = \mathbb{E}_{\mathbf{\Theta}}\left[\mathbf{K}_{il}^{(k)}\right] \tag{3.142}$$

where $\mathbf{K}_{il}^{(k)}$ is modified with random locations and their estimates:

$$\mathbf{K}_{il}^{(k)} \equiv \left[\mathbb{E}[(\hat{X}_i^{(k)} - X_i)(\hat{X}_l^{(k)} - X_l) + (\hat{Y}_i^{(k)} - Y_i)(\hat{Y}_l^{(k)} - Y_l)]\right] / h_{ij} / h_{lj} \tag{3.143}$$

Thus the generic upper bound becomes

$$\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) \le [\exp(b^2)+1]\mathbb{E}_{\mathbf{\Theta}}\left[\sum_{i=1}^{m+n}(a_{ij}^{(k)}\Delta_{ij}^{(k)})^2\right] + \mathbb{E}_{\mathbf{\Theta}}\left[\sum_{i=1}^{m+n}\sum_{l=1,l\neq i}^{m+n} a_{ij}^{(k)}a_{lj}^{(k)}\mathbf{K}_{il}^{(k)}\right]. \tag{3.144}$$

If $a_{ij}^{(k)}$ and $\mathbf{K}_{il}^{(k)}$ are uncorrelated such as the equal weight case,

$$\mathbb{E}_{\mathbf{\Theta}}[a_{ij}^{(k)}a_{lj}^{(k)}\mathbf{K}_{il}^{(k)}] = \mathbb{E}_{\mathbf{\Theta}}[a_{ij}^{(k)}a_{lj}^{(k)}]\mathbb{E}_{\mathbf{\Theta}}[\mathbf{K}_{il}^{(k)}]. \tag{3.145}$$

Moreover, if $\hat{\theta}_i^{(k)}$ and $\hat{\theta}_i^{(k)}$ are assumed to be uncorrelated,

$$\mathbf{K}_{il}^{(k)} = \left[(X_i - \mathbb{E}[\hat{X}_i^{(k)}])(X_l - \mathbb{E}[\hat{X}_l^{(k)}]) + (Y_i - \mathbb{E}[\hat{Y}_i^{(k)}])(Y_l - \mathbb{E}[\hat{Y}_l^{(k)}])\right] / h_{ij} / h_{lj}. \tag{3.146}$$

Then the expectation of $\mathbf{K}_{il}^{(k)}$ over $\boldsymbol{\Theta}$ with the setup of unchanged connection (h_{ij} and h_{lj} keep the same with different realizations of locations):

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\Theta}}[\mathbf{K}_{il}^{(k)}]h_{ij}h_{lj} &= \mathbb{E}_{\boldsymbol{\Theta}}[X_iX_l] + \mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{X}_i^{(k)}]\mathbb{E}[\hat{X}_l^{(k)}]] - \mathbb{E}_{\boldsymbol{\Theta}}[X_i\mathbb{E}[\hat{X}_l^{(k)}]] - \mathbb{E}_{\boldsymbol{\Theta}}[X_l\mathbb{E}[\hat{X}_i^{(k)}]] \\ &\quad + \mathbb{E}_{\boldsymbol{\Theta}}[Y_iY_l] + \mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{Y}_i^{(k)}]\mathbb{E}[\hat{Y}_l^{(k)}]] - \mathbb{E}_{\boldsymbol{\Theta}}[Y_i\mathbb{E}[\hat{Y}_l^{(k)}]] - \mathbb{E}_{\boldsymbol{\Theta}}[Y_l\mathbb{E}[\hat{Y}_i^{(k)}]] \\ &= \mathbb{E}_{\boldsymbol{\Theta}}[X_i]\mathbb{E}_{\boldsymbol{\Theta}}[X_l] + \mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{X}_i^{(k)}]]\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{X}_l^{(k)}]] - \mathbb{E}_{\boldsymbol{\Theta}}[X_i]\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{X}_l^{(k)}]] \\ &\quad - \mathbb{E}_{\boldsymbol{\Theta}}[X_l]\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{X}_i^{(k)}]] + \mathbb{E}_{\boldsymbol{\Theta}}[Y_i]\mathbb{E}_{\boldsymbol{\Theta}}[Y_l] + \mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{Y}_i^{(k)}]]\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{Y}_l^{(k)}]] \\ &\quad - \mathbb{E}_{\boldsymbol{\Theta}}[Y_i]\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{Y}_l^{(k)}]] - \mathbb{E}_{\boldsymbol{\Theta}}[Y_l]\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{Y}_i^{(k)}]] \end{aligned} \tag{3.147}$$

The last equation is valid if θ_i and $\mathbb{E}[\hat{\theta}_l^{(k)}]$ are uncorrelated. The expectation of location estimates with a fixed topology can be written as following by iteratively applied (3.135):

$$\begin{aligned}\mathbb{E}[\hat{\theta}_i^{(k)}] &= \sum_{l_{k-1}=1}^{m+n} a_{l_{k-1}i}^{(k-1)}\mathbb{E}[\hat{\theta}_{l_{k-1}}^{(k-1)}] = \sum_{l_{k-1}=1}^{m+n} a_{l_{k-1}i}^{(k-1)} \sum_{l_{k-2}=1}^{m+n} a_{l_{k-2}l_{k-1}}^{(k-2)}\mathbb{E}[\hat{\theta}_{l_{k-2}}^{(k-2)}] \\ &= \cdots = \sum_{l_{k-1}=1}^{m+n} a_{l_{k-1}i}^{(k-1)} \cdots \sum_{l_0=1}^{m+n} a_{l_0l_1}^{(0)}\theta_{l_0} = \sum_{l_{k-1}=1}^{m+n} \cdots \sum_{l_0=1}^{m+n} a_{l_{k-1}i}^{(k-1)} \cdots a_{l_0l_1}^{(0)}\theta_{l_0} \end{aligned} \tag{3.148}$$

where the last second equality by the fact $\mathbb{E}[\hat{\theta}_{l_0}^{(0)}] = \theta_{l_0}$, which is the anchor's location. Then the expectation over random locations of node connected to node i ($\boldsymbol{\Theta}'$) is

$$\mathbb{E}_{\boldsymbol{\Theta}'}[\mathbb{E}[\hat{\theta}_i^{(k)}]] = \sum_{l_{k-1}=1}^{m+n} \cdots \sum_{l_0=1}^{m+n} \mathbb{E}_{\boldsymbol{\Theta}'}[a_{l_{k-1}i}^{(k-1)} \cdots a_{l_0l_1}^{(0)}]\mathbb{E}_{\boldsymbol{\Theta}'}[\Theta_{l_0}] \tag{3.149}$$

if the weights are uncorrelated to location Θ_{l_0}. $\mathbb{E}_{\boldsymbol{\Theta}'}[\Theta_{l_0}] = \theta_i$ because all uniformly placed anchors connected to node i are inside a circle with centra at θ_i and a prop-

agation radius. Thus

$$\mathbb{E}_{\boldsymbol{\Theta}'}[\mathbb{E}[\hat{\theta}_i^{(k)}]] = \theta_i \mathbb{E}_{\boldsymbol{\Theta}'}\left[\sum_{l_{k-1}=1}^{m+n} \cdots \sum_{l_0=1}^{m+n} a_{l_{k-1}i}^{(k-1)} \cdots a_{l_0 l_1}^{(0)}\right] = \theta_i. \tag{3.150}$$

For the node i as an estimate provider of node j,

$$\mathbb{E}_{\boldsymbol{\Theta}}[\mathbb{E}[\hat{\Theta}_i^{(k)}]] = \mathbb{E}_{\boldsymbol{\Theta}_i}[\mathbb{E}_{\boldsymbol{\Theta}'}[\mathbb{E}[\hat{\Theta}_i^{(k)}]]] = \mathbb{E}_{\boldsymbol{\Theta}_i}[\Theta_i] = \theta_j \tag{3.151}$$

The last equality is because the nodes connecting to the target node j are uniformly distributed with respect to the centra θ_j by setup. Applying the same equality $\mathbb{E}_{\boldsymbol{\Theta}_i}[\Theta_i] = \theta_j$ for other terms in $\mathbb{E}_{\boldsymbol{\Theta}}[\mathbf{K}_{il}^{(k)}]$, it results in

$$\mathbb{E}_{\boldsymbol{\Theta}}[\mathbf{K}_{il}^{(k)}] h_{ij} h_{lj} = 0 \quad \text{and} \quad \mathbb{E}_{\boldsymbol{\Theta}}[a_{ij}^{(k)} a_{lj}^{(k)}] \mathbb{E}_{\boldsymbol{\Theta}}[\mathbf{K}_{il}^{(k)}] = 0. \tag{3.152}$$

Finally, the (generic) upper bound can be simplified under several uncorrelated assumptions and the system of uniformly placed nodes as following:

$$\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) \leq [\exp(b^2) + 1] \mathbb{E}_{\boldsymbol{\Theta}}\left[\sum_{i=1}^{m+n} (a_{ij}^{(k)} \Delta_{ij}^{(k)})^2\right]. \tag{3.153}$$

Then the upper bound can be obtained by substituting the weight $a_{ij}^{(k)}$ with equal weight $h_{ij} / \sum_{i=1}^{m+n} h_{ij}$. If the system is full connectivity, $a_{ij}^{(k)} = 1/(m+n)$ for all i, j, and k, the upper bound is

$$\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) \leq \frac{\exp(b^2) + 1}{(m+n)^2} \mathbb{E}_{\boldsymbol{\Theta}}\left[\sum_{i=1}^{m+n} (\Delta_{ij}^{(k)})^2\right]. \tag{3.154}$$

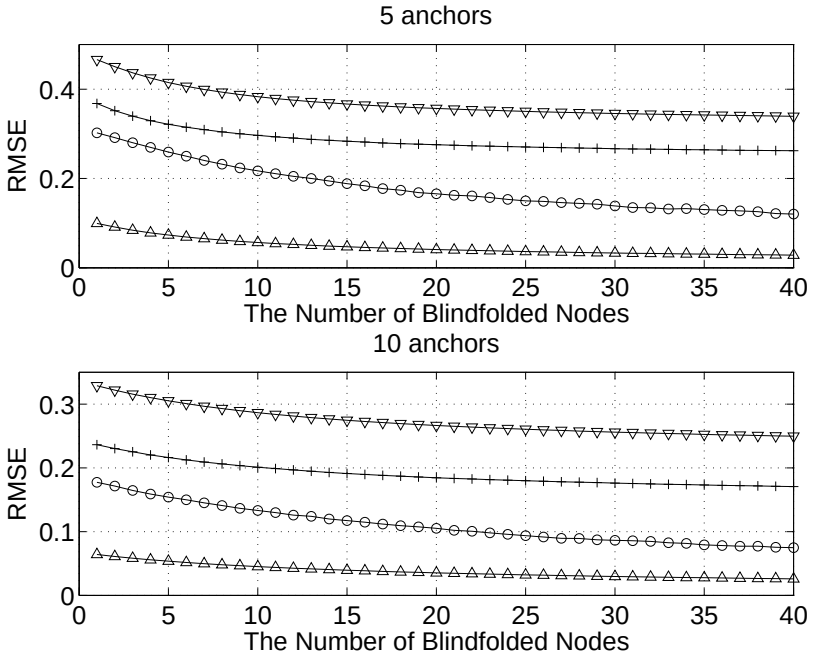

Figure 3.34: RMSE of the LC estimator and its bounds versus the number of randomly placed blindfolded nodes and anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$. The notations are list as o: the estimator, $\triangle$: lower bound, ∇: upper bound (by equal weights), and $+$: approximation.

Moreover, the approximation becomes

$$\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) \leq \mathbb{E}_{\mathbf{\Theta}} \left[\frac{\exp(b^2) + 1}{\sum_{i=1}^{m+n} 1/(\Delta_{ij}^{(k)})^2} \right] \tag{3.155}$$

by using the weights from modified effective distances. Although the above equations require uncorrelated variables, the results with and without correlation are slightly different.

The example with randomly placed nodes inside a unit circle unfortunately in Section 3.4.3 cannot apply to the above bounds for random topology because $\mathbb{E}_{\mathbf{\Theta}_i}[\Theta_i] = \theta_j$ is valid only for $j = 1$ at the origin. Other blindfolded nodes should include more

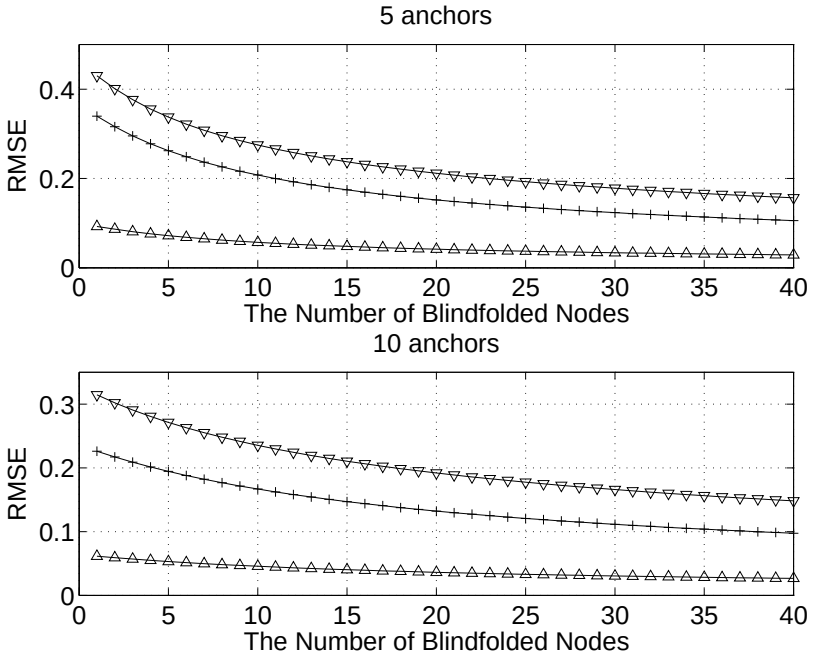

Figure 3.35: RMSE of the bounds for LC estimator versus the number of randomly placed blindfolded nodes and anchors with $n_p = 3$, $\sigma = 5$ and the unit reachable distance. The notations are list as $\triangle$: lower bound, ∇: upper bound (by equal weights), and $+$: approximation.

nodes within its radiation circle to satisfy the equality. Therefore, one has to compute the upper bound and the approximation for each realization of topology as in (3.131) and (3.133) for the fixed topology. Then averaged results over each topology to obtain the upper bound and the approximation. Figure 3.34 presents the RMSE of LC and related bounds in the unit circle case. Both the upper bound and the approximation are assumed the uncorrelated $\mathbf{K}_{il}^{(k)}$. Because of the full connectivity, iterations for all three analyses stop after the second cooperation where $\hat{\theta}_1^{(3)}$ is investigated. The RMSE of the LC estimator is closer to the lower bound as the number of nodes increases. Moreover, the upper bound by equal weight is slightly looser and the approximation is better to clip the worst performance of LC estimator.

Let us consider a larger network where every blindfolded node connects to other uniformly distributed nodes in all directions. In other words, all blindfolded nodes can be viewed to have the same topology as the node 1 in the previous circle example. Thus the example can apply to the simplified lower bound in (3.153) and approximation in (3.155). On the average, the accuracy and penalty are the same for all blindfolded nodes in this network. Investigating the node 1 at the origin can understand the performance of the network. Thus the penalty for each blindfolded node can be replaced by the one of node 1 and only the squares of effective distances to node 1 are required. In other words,

$$(\Omega_{ij}^{(k)})^2 = \frac{\mathrm{Var}(\hat{\omega}_{1i}^{(k)})}{\exp(b^2)+1} + \frac{D_{ij}^2}{h_{ij}} = \frac{1}{\sum_{l=1,l\neq j}^{m+n} \frac{1}{(\Omega_{l1}^{(k-1)})^2}} + \frac{D_{ij}^2}{h_{ij}}, \tag{3.156}$$

$$(\Delta_{ij}^{(k)})^2 = \frac{\mathrm{MSE}(\hat{\theta}_1^{(k)})}{\exp(b^2)+1} + \frac{D_{ij}^2}{h_{ij}} \tag{3.157}$$

for both lower and upper bounds, respectively. The lower bound and approximation then can be obtained by computing $\mathbb{E}\left[\frac{1}{\sum_{i=1}^{m+n} 1/\Omega_{ij}^2}\right]$ and $\mathbb{E}\left[\frac{1}{\sum_{i=1}^{m+n} 1/\Delta_{ij}^2}\right]$ through simple simulations. The upper bound by the equal weight is even easier. Because $\mathbb{E}_D\left[D_{ij}^2\right] = r^2/2 = 1/2$ where r is the radius of a circle, the upper bound with equal weight can be simplified as

$$\mathrm{MSE}(\hat{\theta}_j^{(k+1)}) \leq \frac{\exp(b^2)+1}{2(m+n)} + \frac{m\mathrm{MSE}(\hat{\theta}_1^{(k)})}{(m+n)^2} \tag{3.158}$$

in full connectivity situations.

The resulting bounds are presented in Figure 3.35 and the upper bound and the approximation are much lower than the ones in in Figure 3.34. Again, iterations stop after the second cooperation where $\hat{\theta}_1^{(3)}$ is investigated. As a result, all three bounds in a large uniformly distributed network provide quantities to quickly understand the accuracy of linear combination estimator.

In conclusion, lower and upper bounds depend on the effective distances and scaling constants $\exp(b^2) - 1$ and $\exp(b^2) + 1$. Thus knowing $\mathbb{E}\left[\frac{1}{\sum_{i=1}^{m+n} 1/\Omega_{ij}^2}\right]$ can quantify the best accuracy of the cooperative LC estimator, which is similar to know the $\mathbb{E}\left[\frac{1}{\sum_{i=1}^{n} 1/D_i^2}\right]$ in the non-cooperative case. The worst accuracy is controlled $\mathbb{E}\left[\frac{1}{\sum_{i=1}^{m+n} 1/\Delta_{ij}^2}\right]$ under the assumption of few uncorrelated results. Again the lower bound, upper bound, and the appropriation provide a quick view of accuracy. Similar to the non-cooperative case, these bounds are especially important when the topologies are random. It therefore does not needs to run many trails for each topology to obtain MSE with respect to noisy power measurements.

3.6 Computation and Communication Costs

This section discusses both computation and communication costs for three distributed methods: the proposed LC, the MLE by coordinate descent, and dwMDS. The full connectivity where all m blindfolded nodes and n anchors can hear one another in the network is assumed to investigate the highest cost. Because all three methods require iterations, K denotes the number of iterations.

3.6.1 Computation Cost

The time complexity for the cooperative linear combination estimator is obtained by checking each step. Because of the complexity in computing directions, effective distances, combining weights and linear combination are all $O((m+n)K)$ for each blindfolded node, the time complexity for each blindfolded node in LC is $O((m+n)K)$ and the complexity for the whole network is $O(m(m+n)K)$. If a blindfolded node in LC excludes the information from its receiver e.g., the analysis part and LCa, it is required to compute different $\hat{\omega}_{ij}^{(k)}$ in (3.14) and/or $\gamma_{ij}^{(k-1)}$ in (3.33) for $i = 1, \ldots, m$ and $j = 1, \ldots, m$. However, computation can be reduced by one summation and $m-1$ subtractions instead of m summations as in (3.40). Thus the time complexity

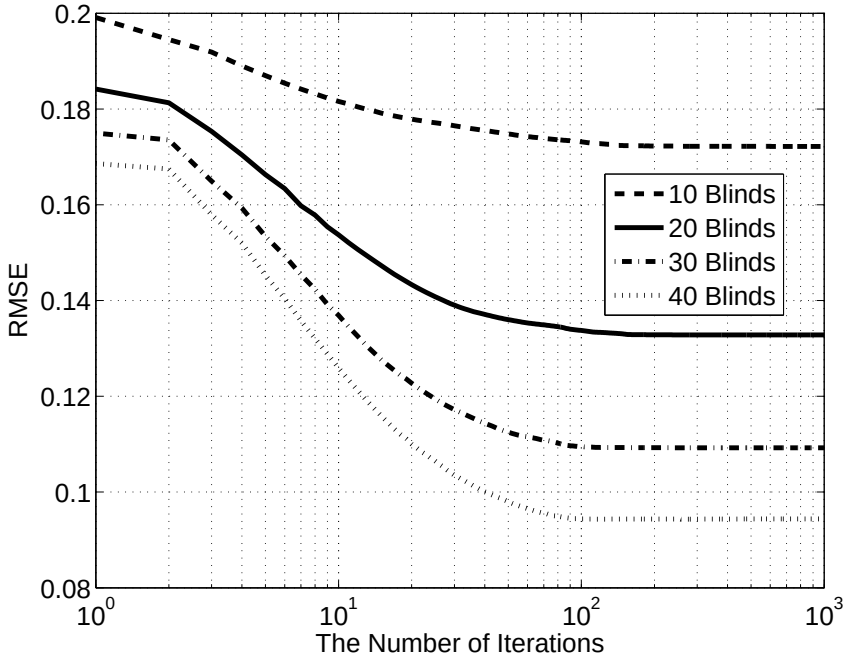

Figure 3.36: The RMSE of LC versus the number of iterations for randomly placed blindfolded nodes and 7 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

is still $O((m+n)K)$ for each node. Full calibration (fc) and partial calibration (pc) can be viewed as LC with more iterations. Thus their complexity is $O((m+n)KL)$ where L is the number of inner iterations. The average L for the unit circle case is less than 10. The extra cost is acceptable.

Since K is crucial to time complexity, Figures 3.36 and 3.37 present how the RMSE of LC changes with iterations by adding randomly placed anchors and blindfolded nodes inside a unit circle. As mentioned before in Section 3.4.3, the RMSE investigated here is the blindfolded node at the origin. The estimates of LC are stable after hundreds of iterations and the decrease of RMSE is not monotonic.

For the MLE using coordinate descent (CD), both finding the minimum for each coordinate and the number of iterations are the keys for time complexity. Assume

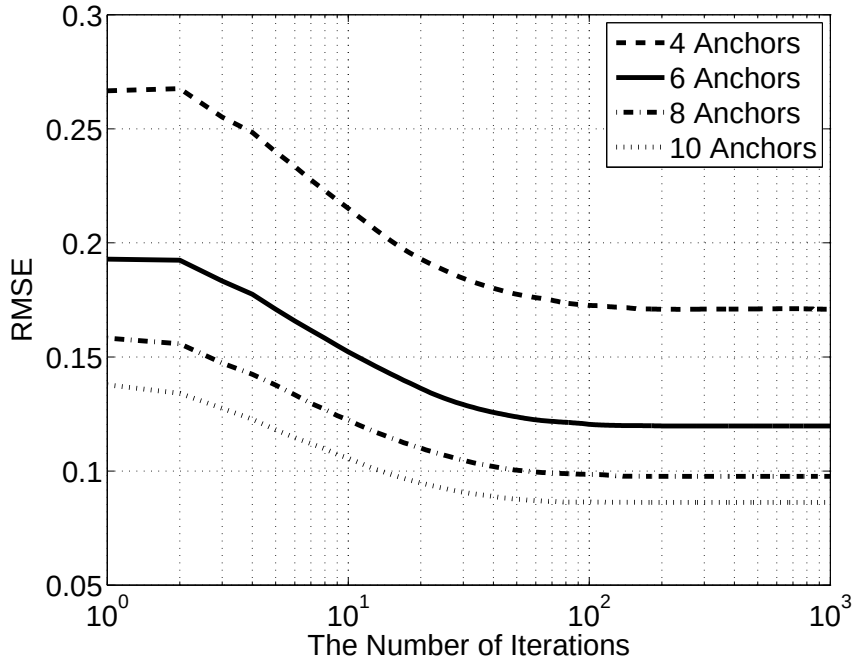

Figure 3.37: The RMSE of LC versus the number of iterations for randomly placed anchors and 30 blindfolded nodes inside a unit circle with $n_p = 3$ and $\sigma = 5$.

all coordinates have the same size of candidate, i.e., $|\mathcal{S}_{x_i}| = |\mathcal{S}_{y_i}| = |\mathcal{S}|$ for $i = 1, \ldots, n$ and let K be the number of iterations. Because searching for a minimum is linear time, the complexity of CD for a blindfolded in each iteration is linear in the number of nodes connected to a blindfolded node and the size of search space, i.e., $O((m+n)|\mathcal{S}|)$. After including the iterations and all m blindfolded nodes, time complexity of CD is $O(m(m+n)K|\mathcal{S}|)$ for the network and it is $O((m+n)K|\mathcal{S}|)$ for each blindfolded node.

The computational time is dominated by the size $|\mathcal{S}|$ because the searching size is usually much larger than the number of connected nodes and iterations. Understanding how the search size affects RMSE is essential for an exhaustive search. Figure 3.38 shows the RMSE for different step sizes where 0.005 is the default step

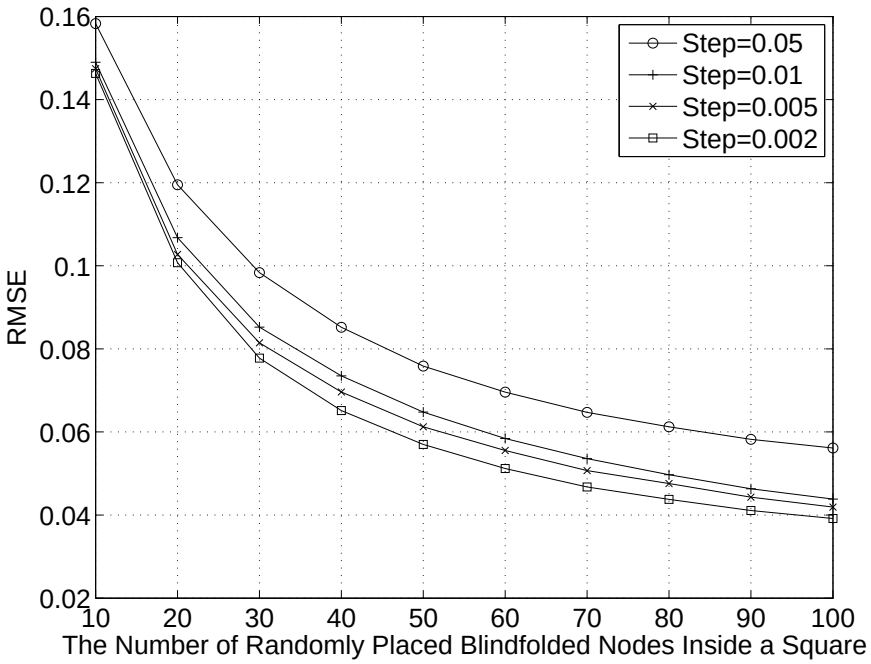

Figure 3.38: The RMSE of MLE:CD with different search steps for randomly placed blindfolded nodes inside a unit square with 4 corner anchors under $n_p = 3$ and $\sigma = 5$.

in this dissertation. The RMSE monotonically decreases as the step decreases in the figure. Mimicking the definition of relative error, the relative difference of RMSE is defined as in (2.74) to quantify the accuracy gap between two step sizes:

$$\delta\text{RMSE} \equiv \frac{\text{RMSE} - \text{RMSE}_0}{\text{RMSE}_0} \tag{3.159}$$

where RMSE_0 is the reference RMSE with a small step size. The relative RMSE with the RMSE by step size 0.002 as RMSE_0 is shown in Figure 3.39. The maximum relative RMSE is 8.7% when using the default step size 0.005. Figure 3.40 shows the average iterations for various step sizes.

For dwMDS, each blindfolded node should have the complexity $O((m+n)K)$ and the complexity for the whole network is $O(m(m+n)K)$. Both are the same as the

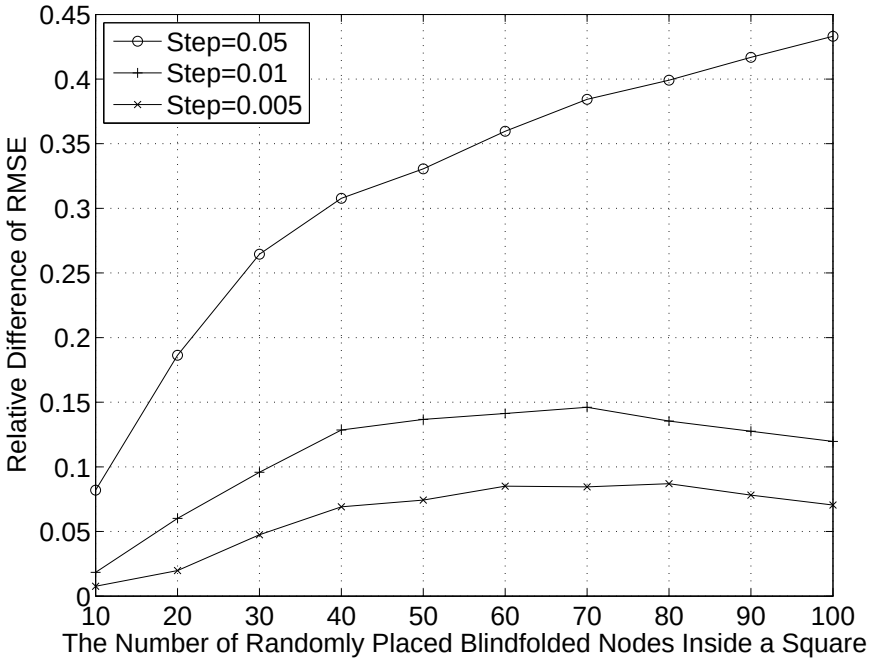

Figure 3.39: The relative difference of RMSE for MLE:CD where RMSE_0 has the step size 0.002 for randomly placed blindfolded nodes inside a unit square with 4 corner anchors under $n_p = 3$ and $\sigma = 5$.

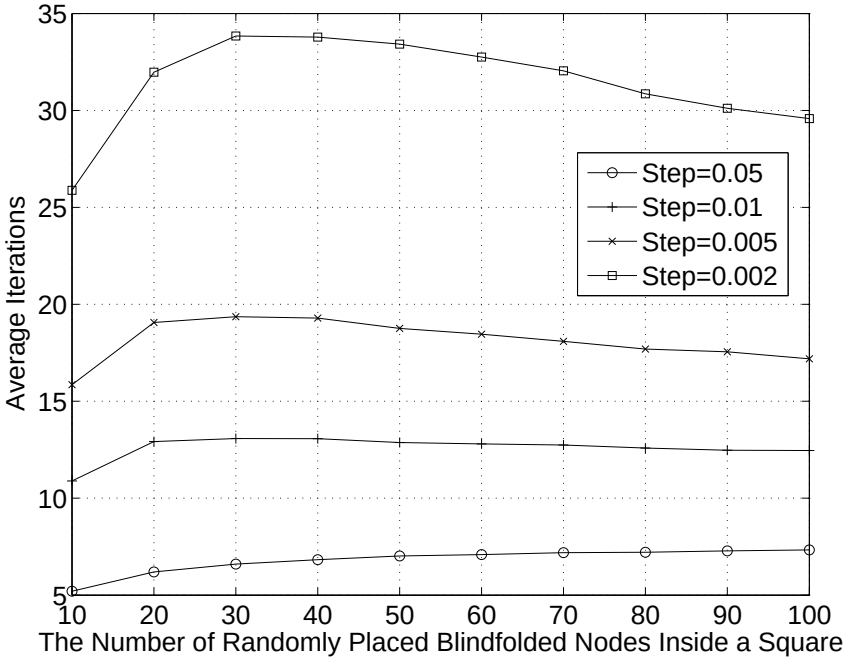

Figure 3.40: Average number of the MLE:CD's iterations for randomly placed blindfolded nodes inside a unit square with 4 corner anchors under $n_p = 3$ and $\sigma = 5$.

LC.

The average number of iterations has been investigated in Section 3.4 and the results depend the stopping criterion ϵ. It is also important to check the distribution of iterations to understand how K affect the computational cost and when to stop the iteration. Two illustrated results are shown in Figures 3.41 and 3.42. Two successive method: LCn-pc and MLE: CD reach stable region fastest before 100 iterations; LCn-pc is even slightly better than MLE: CD. dwMDS is slightly slower than two successive ones. LC is slowest but should finish iterations before 200 iterations.

In summary, the complexity for each blindfolded node is linear with the number of blindfolded nodes m and the complexity for the whole network is quadratic with the number of blindfolded nodes in all three compared methods. However, checking

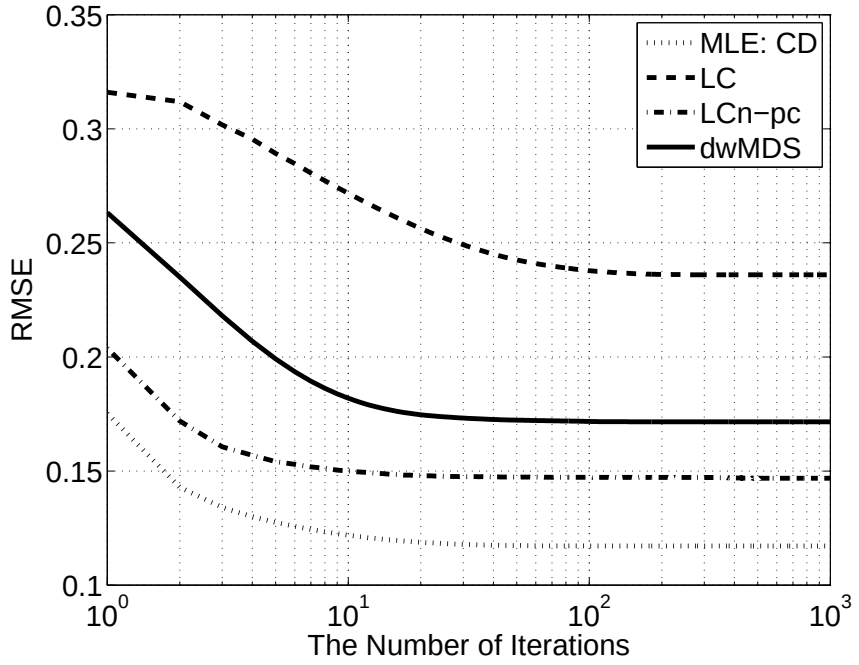

Figure 3.41: The RMSE versus the number of iterations for randomly placed 7 anchors and 30 blindfolded nodes inside a unit circle with $n_p = 3$ and $\sigma = 5$.

the running time can explicitly reveal how fast they are executing. The running time are calculated by the function `cputime` of Matlab as in the case of non-cooperative localization. Matlab (R2007b) is installed on a personal computer equipped with an Intel Xeon 2.66 GHz CPU and 4 GB RAM; the OS is Linux version 2.6.26-2-amd64 (Debian). The computation time under 10^3 trails with randomly placed blindfolded nodes inside a unit square are presented in Figure 3.43. The MLE:CD requires the most time even though it requires the fewest iterations. dwMDS has the least time, roughly 30 times faster than the MLE:CD. All three LCs have similar running time because the original LC requires most iterations. To exclude the influence of iterations, the running time divided by the average iterations are shown in Figure 3.44. The dwMDS is still fastest and LC is the second fastest. LC-fc and LCn-pc

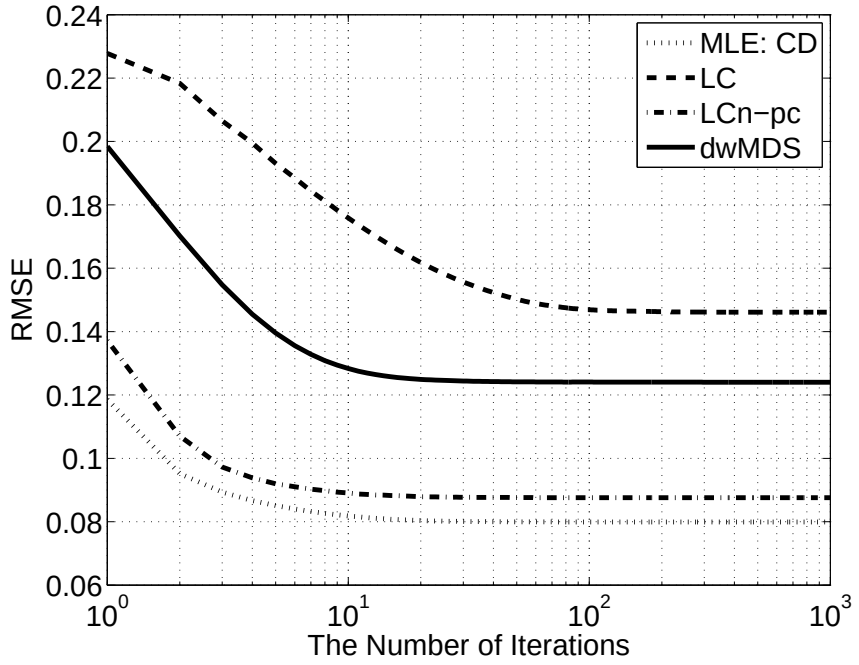

Figure 3.42: The RMSE versus the number of iterations for randomly placed 10 anchors and 40 blindfolded nodes inside a unit circle with $n_p = 3$ and $\sigma = 5$.

have almost the same running time.

3.6.2 Communication Cost

In contrast to the computation cost focusing on node self cost, the communication cost investigates the inter-node cost from exchanging information. Then the communication cost mainly depends on the amount of exchanged information, the number of iterations, and the transmitting power in all three discussed algorithms. Because the communication occurs when one node connects to its neighbors, the total energy cost in the network can be expressed as the multiplication of number of packets transmitted and average amount of energy to ensure one packet transmitted [59].

The number of transmitted packets is discussed as follows. For an iteration in the LC algorithm without extraction, a blindfolded node needs to send its estimated

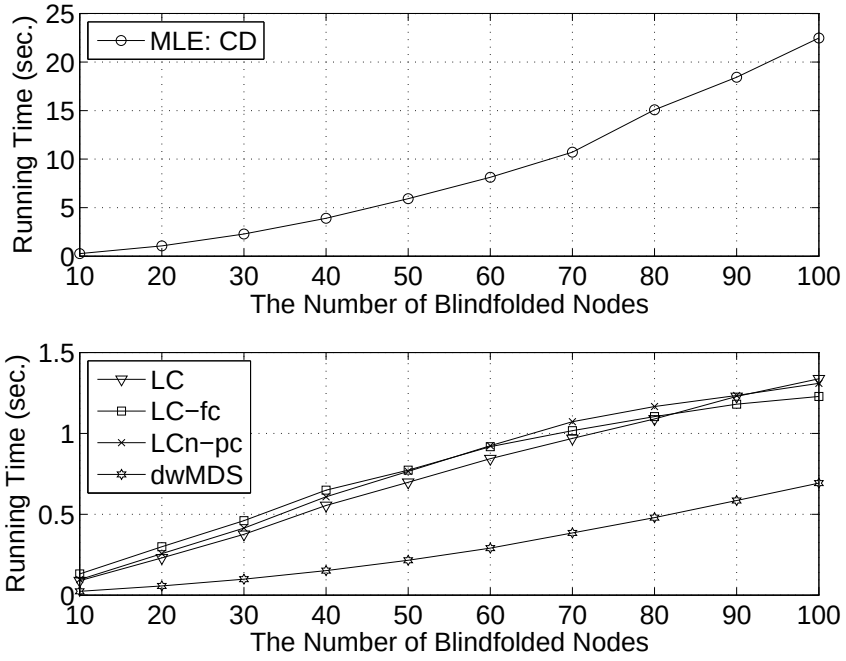

Figure 3.43: Computation time of various localization methods for randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$ under 10^3 trails.

location $\hat{\theta}_j^{(k+1)}$, two stopping indicators $S^{(k)}$ and $N^{(k)}$, and penalty $\hat{\gamma}_j^{(k)}$, as shown in the step 11 of the LC estimator in Section 3.3.2. On the other hand, the one time and only transmitted information of an anchors is its position θ_j. Thus the total packets transmitted in the 2D LC network are $5nK + 2m$ where K is the number of iterations. If the LC with weight extraction is applied, the penalty term becomes $\hat{\gamma}_{ji}^{(k)}$ and a blindfolded node is required to transmit at most $n-1$ different $\hat{\gamma}_{ji}^{(k)}$ to the network as the algorithm suggests. However, the penalty can be computed at the receiver i by $\hat{\eta}_{ji}^{(k)} = \hat{\eta}_j^{(k)} - \frac{1}{(\hat{\delta}_{ij}^{(k)})^2}$ because the receiver knows $\hat{\delta}_{ij}^{(k)}$. Thus node j needs to broadcast only $\eta_j^{(k)}$ and the total transmitted packets are still $5nK + 2m$ in the 2D case. Thus the packet costs are $O(nK + m)$ for the LC estimator without and with weight extraction. For the MLE by coordinate descent (CD) methods, the

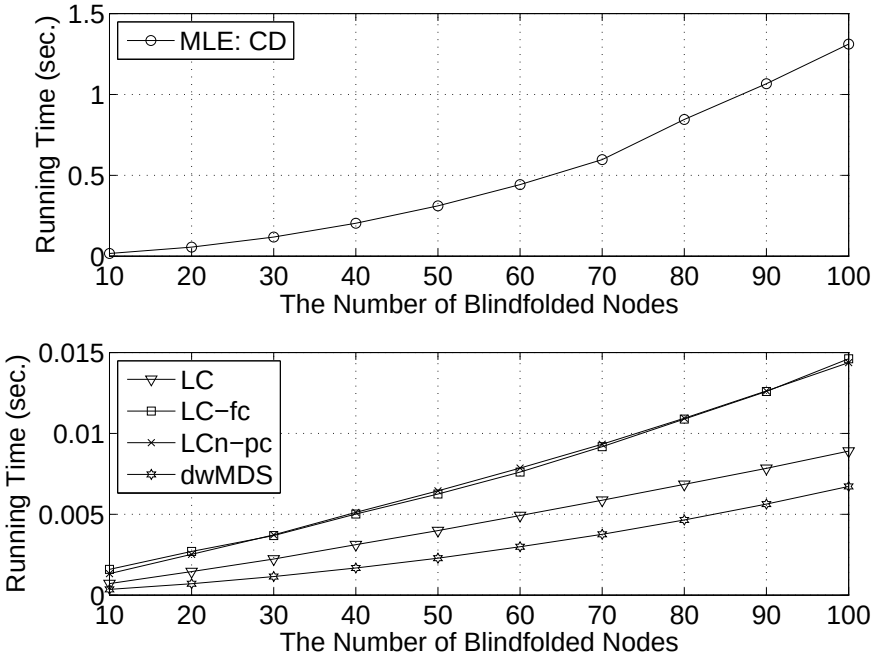

Figure 3.44: Average one iterated computation time of various localization methods for randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

transmitted packets for a blindfolded node are $\hat{\theta}_j^{(k+1)}$, $S^{(k)}$, and $N^{(k)}$. Thus the total packets for CD in 2D are $4nK + 2m$, which is slightly less than the cost of a LC estimator. In the dwMDS, each blindfolded requires to send estimated locations and a global objective function to be minimized. The total packets for the dwMDS in 2D are $3nK + 2m$, which is slightly less than the CD and the LC estimator. Again, the number of transmitted packets for both the CD and the dwMDS are $O(nK + m)$, the same as for the LC estimator.

The relationship between transmitting power (P_{Tx}) and received power (P_{Rx}) is expressed as

$$P_{\mathrm{Rx}} = P_{\mathrm{Tx}} P_{\mathrm{L}}(d) \tag{3.160}$$

where $P_{\mathrm{L}}(d)$ denotes the power loss at distance d and is modeled as the log-normal

shadowing as in (2.2):

$$P_{\mathrm{L}}(d) = p_0 \left(\frac{d}{d_0}\right)^{-n_p} 10^{\frac{Z}{10}}. \tag{3.161}$$

To ensure the connection, the received power should be at least p_{th}. For two transmitters Tx1 and Tx2 at different distances d_1 and d_2, respectively, to ensure a received power p_{th}, the equality is given by

$$p_{\mathrm{th}} = P_{\mathrm{Tx1}} p_0 \left(\frac{d_1}{d_0}\right)^{-n_p} 10^{\frac{Z_1}{10}} = P_{\mathrm{Tx2}} p_0 \left(\frac{d_2}{d_0}\right)^{-n_p} 10^{\frac{Z_2}{10}}. \tag{3.162}$$

Then the ratio of two transmitting powers is

$$\frac{P_{\mathrm{Tx1}}}{P_{\mathrm{Tx2}}} = \left(\frac{d_2}{d_1}\right)^{-n_p} 10^{\frac{Z_2 - Z_1}{10}}. \tag{3.163}$$

The systems designed by averaging the fluctuation of powers has the ratio of average transmitting powers

$$\frac{P_{\mathrm{Tx1}}}{P_{\mathrm{Tx2}}} = \left(\frac{d_2}{d_1}\right)^{-n_p} = \left(\frac{d_1}{d_2}\right)^{n_p}. \tag{3.164}$$

The comparison of power usage in the partial connectivity example in Section 3.4.2 can be computed by the above ratio. In the environment given $n_p = 3$, the transmitting power of $d_{\mathrm{th}} = 2$ (threshold distance) is 15 times of the one of $d_{\mathrm{th}} = 0.4$.

Because the powers to transmit a packet are the same for all three algorithms when the system is set, the number of transmitted packets determines the communication costs. The number of transmitted of all three algorithms are slightly different and are with the complexity $O(nK + m)$. Thus the number of iterations K which represents inter-node communications dominates the cost of communications. The MLE:CD has the lowest power consumption in the examples among this dissertation.

3.7 Convergence Analysis

Unlike dwMDS, the convergence of the cooperative LC family is not guaranteed because it does not satisfy all fixed-point theorems mentioned in the non-cooperative LC case in Section 2.7. The simulations are used to check the convergence for LCs under several conditions in Section 3.4. The original LC converges (having fixed points) in almost all the running simulations with the exception of 0.01% unfixed point in a few of unit circle topologies. LC-fc may not converge especially in a sparse network and a low SNR environment. In addition, LCn-pc is convergent in most cases.

If a LC estimator obtains a fixed point, it is desirable to check if the fixed point is stable by observing its spectral radius. The 2D case of the original LC having m blindfolded nodes and n anchors is denoted by

$$\begin{aligned}
\hat{x}_j^{(k+1)} &\equiv f_j(\hat{x}_1^{(k)}, \hat{y}_1^{(k)}, \ldots, \hat{x}_{m+n}^{(k)}, \hat{y}_{m+n}^{(k)}) \\
&= \sum_{i=1}^{m+n} a_{ij}^{(k)} \left(\hat{x}_i^{(k)} + \hat{d}_{ij} \frac{\hat{x}_j^{(k)} - \hat{x}_i^{(k)}}{\sqrt{(\hat{x}_j^{(k)} - \hat{x}_i^{(k)})^2 + (\hat{y}_j^{(k)} - \hat{y}_i^{(k)})^2}} \right) \\
&\equiv \sum_{i=1}^{m+n} a_{ij}^{(k)} f_{ij}(\hat{x}_i^{(k)}, \hat{y}_i^{(k)}, \hat{x}_j^{(k)}, \hat{y}_j^{(k)}) \qquad (3.165) \\
\hat{y}_j^{(k+1)} &\equiv g_j(\hat{x}_1^{(k)}, \hat{y}_1^{(k)}, \ldots, \hat{x}_{m+n}^{(k)}, \hat{y}_{m+n}^{(k)}) \\
&= \sum_{i=1}^{m+n} a_{ij}^{(k)} \left(\hat{y}_i^{(k)} + \hat{d}_{ij} \frac{\hat{y}_j^{(k)} - \hat{y}_i^{(k)}}{\sqrt{(\hat{x}_j^{(k)} - \hat{x}_i^{(k)})^2 + (\hat{y}_j^{(k)} - \hat{y}_i^{(k)})^2}} \right) \\
&\equiv \sum_{i=1}^{m+n} a_{ij}^{(k)} g_{ij}(\hat{x}_i^{(k)}, \hat{y}_i^{(k)}, \hat{x}_j^{(k)}, \hat{y}_j^{(k)}) \qquad (3.166)
\end{aligned}$$

for $j = 1, \ldots, m$. It should be noticed that $\hat{x}_i^{(k)} = x_i$ and $\hat{y}_i^{(k)} = y_i$ for $i = m + 1, \ldots, m + n$, the anchors' locations.

Let $\mathbf{T}$ denote the iterative function. When a fixed point exists, i.e., $\mathbf{T}(\boldsymbol{\theta}^*) = \boldsymbol{\theta}^* = [x_1^*, y_1^*, \ldots, x_m^*, y_m^*]$ where $\mathbf{T}$ represents the iterative LC algorithm. The Jacobian of $\mathbf{T}(\boldsymbol{\theta})$ is

$$\mathbf{T}'(\boldsymbol{\theta}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial x_m} & \frac{\partial f_1}{\partial y_m} \\ \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial y_1} & \cdots & \frac{\partial g_1}{\partial x_m} & \frac{\partial g_1}{\partial y_m} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial y_1} & \cdots & \frac{\partial f_m}{\partial x_m} & \frac{\partial f_m}{\partial y_m} \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial y_1} & \cdots & \frac{\partial g_m}{\partial x_m} & \frac{\partial g_m}{\partial y_m} \end{bmatrix} \tag{3.167}$$

Here $\boldsymbol{\theta} = [x_1, y_1, \ldots, x_m, y_m]$ represents $\boldsymbol{\theta}^*$ to simplify notations. Similarly, let weight $a_{ij}^{(k)} = a_{ij}^*$ for the fixed points and $a_{ij} = a_{ij}^*$ for simplification. Assume that a_{ij} is independent of location estimates. Thus the elements in the Jacobian are as follows:

$$\begin{aligned} \frac{\partial f_j(\boldsymbol{\theta})}{\partial x_j} &= \sum_{i=1}^{m+n} a_{ij}\hat{d}_{ij} \frac{1}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \left[1 - \frac{(x - x_i)^2}{(x_j - x_i)^2 + (y_j - y_i)^2}\right] \\ &= \sum_{i=1}^{m+n} a_{ij}\hat{d}_{ij} \frac{(y_j - y_i)^2}{[(x_j - x_i)^2 + (y_j - y_i)^2]^{3/2}} \qquad (3.168) \\ \frac{\partial g_j(\boldsymbol{\theta})}{\partial y_j} &= \sum_{i=1}^{m+n} a_{ij}\hat{d}_{ij} \frac{(x_j - x_i)^2}{[(x_j - x_i)^2 + (y_j - y_i)^2]^{3/2}} \qquad (3.169) \\ \frac{\partial f_j(\boldsymbol{\theta})}{\partial y_j} &= -\sum_{i=1}^{m+n} a_{ij}\hat{d}_{ij} \frac{(x_j - x_i)(y_j - y_i)}{[(x_j - x_i)^2 + (y_j - y_i)^2]^{3/2}} \;=\; \frac{\partial g_j(\boldsymbol{\theta})}{\partial x_j} \qquad (3.170) \\ \frac{\partial f_j(\boldsymbol{\theta})}{\partial x_i} &= a_{ij}\left[1 - \hat{d}_{ij} \frac{(y_j - y_i)^2}{[(x_j - x_i)^2 + (y_j - y_i)^2]^{3/2}}\right] \qquad (3.171) \\ \frac{\partial g_j(\boldsymbol{\theta})}{\partial y_i} &= a_{ij}\left[1 - \hat{d}_{ij} \frac{(x_j - x_i)^2}{[(x_j - x_i)^2 + (y_j - y_i)^2]^{3/2}}\right] \qquad (3.172) \\ \frac{\partial f_j(\boldsymbol{\theta})}{\partial y_i} &= a_{ij}\hat{d}_{ij} \frac{(x_j - x_i)(y_j - y_i)}{[(x_j - x_i)^2 + (y_j - y_i)^2]^{3/2}} \;=\; \frac{\partial g_j(\boldsymbol{\theta})}{\partial x_i} \qquad (3.173) \end{aligned}$$

for $i, j = 1, \ldots, m$. If there is no connection between node i and node j ($h_{ij} = 0$), the corresponding $a_{ij} = 0$ and the terms containing i and j in the above equations are also 0.

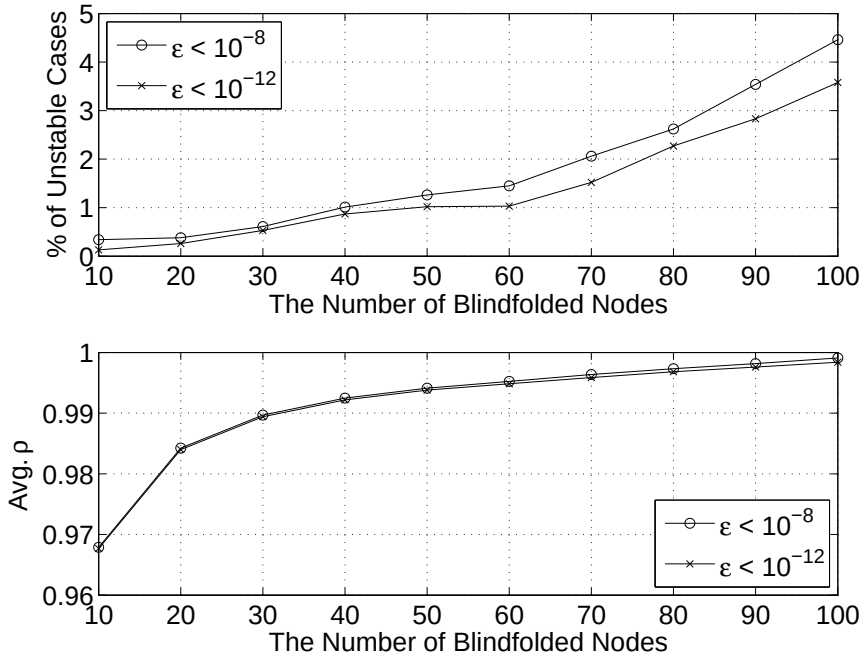

Figure 3.45: The stability of the LC estimator inside a unit square with $n_p = 3$ and $\sigma = 5$.

The spectral radius of $\mathbf{T}'(\theta)$ is defined as

$$\rho(\mathbf{T}'(\theta)) \equiv \max(\{|\lambda_1|, \ldots, |\lambda_m|\}) \tag{3.174}$$

where λ_i is an eigenvalue of $\mathbf{T}'(\theta)$. Thus if $\rho(\mathbf{T}'(\theta)) < 1$, convergence is ensured.

Figure 3.45 illustrates the percentage of unstable cases and average spectral radius with full connectivity inside a unit square as in Section 3.4.1. It shows that there are more unstable estimates as the number of blindfolded nodes increases. More unstable estimates also make the average spectral radius larger. Moreover, the smaller stopping criterion $\epsilon = 10^{-12}$ has fewer unstable points than the one with $\epsilon = 10^{-8}$ which indicates the estimates stops at $\epsilon = 10^{-8}$ may not provide true fixed points.

3.8 Connectivity versus Cost

Although more connections generally provides better estimated positions, it results in more power consumptions and the decrease of estimation error RMSE can be saturated after certain level of connectivity. This section explores the numbers or density of connected nodes (either anchors or blindfolded nodes) to a blindfolded node to ensure the performance such as positioning accuracy within a desired range. It is also recognized that estimation accuracy and transmitting power are trade-offs in localizations. Because connecting more node can be achieved by adding more nodes or increasing power, the lowest cost system can be obtained by knowing the cost ratio of device and energy. Finally, the effect of adding blindfolded nodes instead of anchors is answered at the end of this section.

3.8.1 Connected Node versus Estimation Error

The analytical and numerical results inside a unit square and circle presented in Sections 3.5 and 3.4, respectively, can be used to understand the relationship between the number of connected nodes and estimation error. Because the RMSE results can be scaled by the size of observation area, system designers can find the required nodes to ensure the desired locating error from the above unit topologies.

The simulations in Section 3.4.1 have shown how the RMSE decreases versus the number of blindfolded nodes in a unit square under full connectivity. Results of partial connectivity with many nodes in a network are presented in Section 3.4.2 which provides another point of view on connectivity. Figure 3.46 illustrates how different total blindfolded nodes affect positioning error as the transmitting power increases. Increasing d_{th} from 0.4 to 0.6 and from 0.8 to 1.2 both make the 27/8 times more power consumptions for $n_p = 3$. However, the RMSE with total 60 blindfolded nodes decrease 0.0490 and 0.0203, respectively. This shows that increasing power is

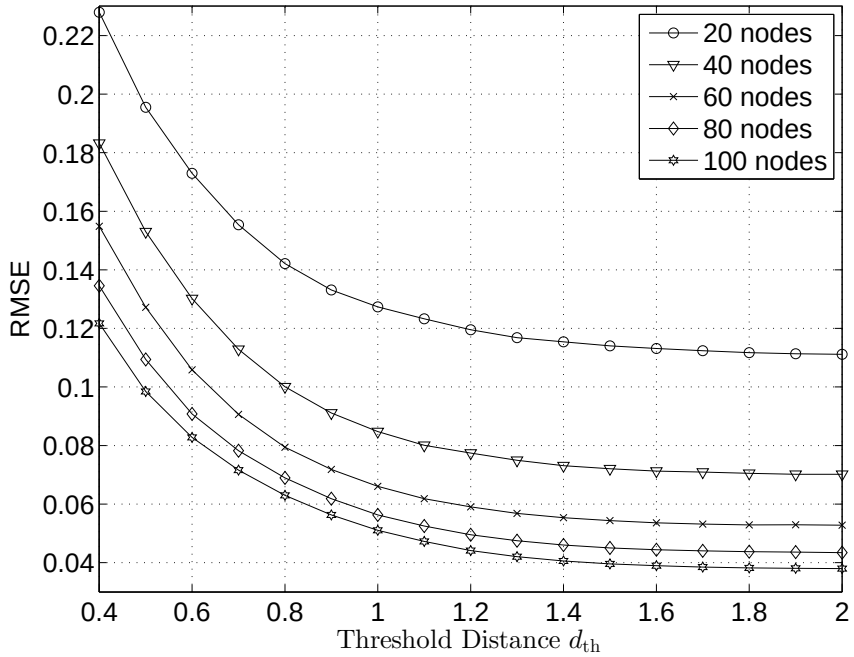

Figure 3.46: RMSE of LCn-pc versus threshold distance given various randomly placed blindfolded nodes inside a unit square with $n_p = 3$ and $\sigma = 5$.

more efficient in low power and the improvement of accuracy saturates. To achieve RMSE = 0.1, one can choose from 40-node to 100-node systems. Then the lowest cost system should consider both the cost of wireless device and transmitting power (battery). The number of connected nodes can be referred to in Figures 3.16 and 3.17.

Similarly, the decrease of RMSE for all blindfolded nodes versus the number of anchors and blindfolded node inside a unit circle are presented in Section 3.4.3. To investigate how adding extra anchors or blindfolded nodes affects the accuracy, a single blindfolded node placed at the origin with transmitting radius 1 (i.e., the reachable distance 1) for all nodes in the network. Thus all nodes inside the unit circle with centre at the origin can be heard by the particular blindfolded node at

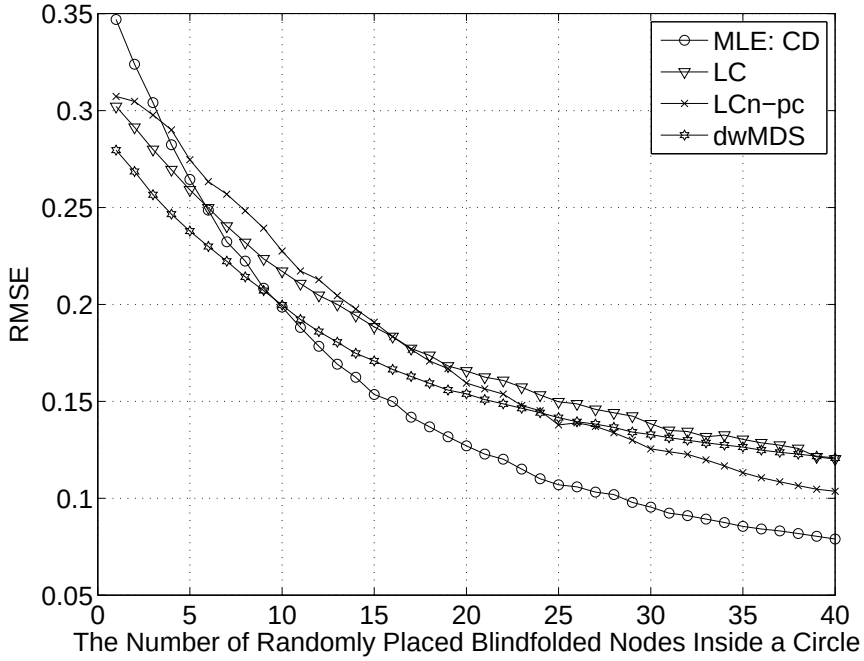

Figure 3.47: RMSE of various methods by adding the number of randomly placed blindfolded nodes and 5 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

the origin. The results are illustrated in Figures 3.47 and 3.48.

3.8.2 Spatial Poisson Point Process

Instead of the number of placed nodes, the node density provides another approach to connectivity. Consider a homogeneous spatial Poisson point process where the density is fixed. Let λ be the expected number of nodes in a unit area. Also let r denote the maximum transmitting range of for all nodes. Thus the signal of the nodes inside the circle πr^2 can be received by the blindfolded node at the centre. The probability of N random nodes existing in the above region can be written as

$$\Pr(N = n) = \frac{(\lambda \pi r^2)^n}{n!} e^{-\lambda \pi r^2}. \tag{3.175}$$

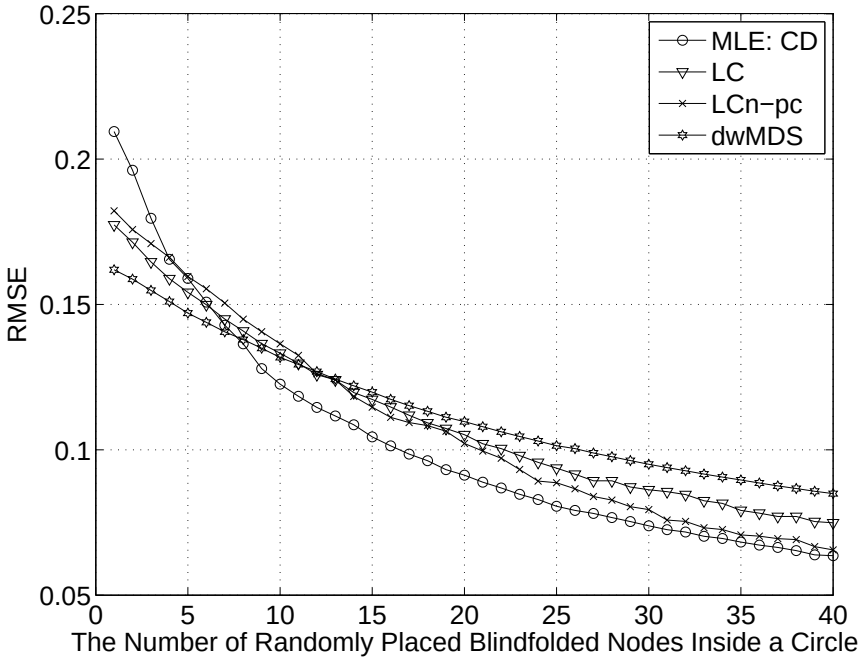

Figure 3.48: RMSE of various methods by adding the number of randomly placed blindfolded nodes and 10 anchors inside a unit circle with $n_p = 3$ and $\sigma = 5$.

The locations of nodes are placed according to a uniform distribution in any observation area. Moreover, let d_i represent the distance between random node i and the blindfolded node. Knowing $N = n$ nodes exist in the observed region, the distribution of $d_1^2, d_2^2, \ldots d_n^2$ is the same as the ordered statistics of n random samples from 0 to r^2. In summary, the setup of a spatial Poisson point process can use the results from the unit circle and the ordered statistics can provide the analytical results from the upper and lower bounds of the linear combination estimator.

3.8.2.1 Density versus Transmitting Range

The number of connected nodes increases when either the density λ or the maximum transmitting range r increases. To understand how two different approaches affect the system design, a non-cooperative example is considered as follows. A

blindfolded node is placed at the origin and it desires to obtain the RMSE = 0.1 under $n_p/\sigma = 3/5$ by using the signals and locations of anchors. Given a fixed r, the minimal density of anchors λ_a is the smallest λ which satisfies

$$\sum_{n=3}^{\infty} \Pr(N = n|N \geq 3)\text{MSE}(r, n) \quad \text{subject to} \quad \Pr(N \geq 3) \geq 0.99 \tag{3.176}$$

where

$$\Pr(N = n|N \geq 3) = \frac{\Pr(N = n)}{\sum_{n=3}^{\infty} \Pr(N = n)} \tag{3.177}$$

is a function of λ and r. MSE(r, n) is computed by the r^2 times the simulated MSE of the LC estimator with unit circle ($r = 1$) in Section 2.5 because the MSE is scaled with the area in RSS localization. The lower bound and upper approximation use the same approach with exception of $\mathbb{E}\left[\frac{1}{\sum_{j=1}^{n} 1/D_j^2}\right]$ and scales $\exp(b^2)-1$ and $\exp(b^2)+1$, respectively. It should remember that the quantity used for bounds can be computed by the ordered statistics of a uniform distribution.

The resulting pairs of (r, λ_a) to achieve the target accuracy are shown in Figure 3.49. The expected minimal number of anchors ($\lambda_a \pi r^2$) inside the circle versus the transmitting radius r are presented in Figure 3.50. The transmitting power for an anchor is assumed to be 1 when $r = 1$ and the total expected power consumption versus r are also shown in Figure 3.50. It is also found that both the number connected anchors and total power consumption monotonically increase as r becomes larger. Thus the system with high density and small transmitting is preferred as long as the constraint $\Pr(N \geq 3) \geq 0.99$ is satisfied.

Another point of view is to improve the position accuracy of an existing system. Let consider a large system with a fixed transmitting radius $r = 1$ for all nodes and density of anchors $\lambda_a = 6.2$. The corresponding expected number of anchors con-

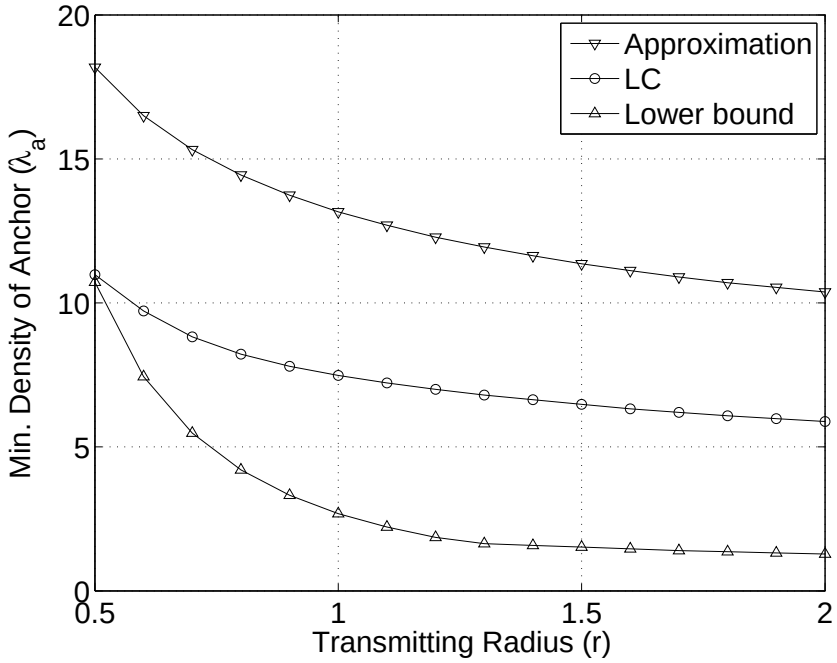

Figure 3.49: Transmitting radius versus minimal density of anchors with desired RMSE=0.1 for LC methods inside a unit circle with $n_p = 3$ and $\sigma = 5$.

nected to the blindfolded nodes is 19.47 and the total expected power consumption is 19.47 inside the circle. To achieve the RMSE = 0.1, one can increase r to 1.7, or λ_a to 7.48, or any other (r, λ) pairs on the curve. The expected number of anchors for $(r, \lambda_a) = (1, 7.48)$ is 23.50. To increase λ_a, the system requires to add about 4 extra anchors on average and the total extra power consumption is 4. On the other hand, increasing r requires no extra added anchor if the large network is assumed to be large enough. The expected number of connected anchors are 56.29 now which clearly means the more computations than the other one with 23.50 nodes. Moreover, the extra power $1.7^{n_p} - 1^{n_p} = 3.91$ for each anchors outside the unit circle as discussed in (3.164) and this results in total extra power cost is $3.91 \times (56.29 - 19.47) = 143.97$ which is much larger than the $(r, \lambda_a) = (1, 7.48)$ system. However, the relative cost

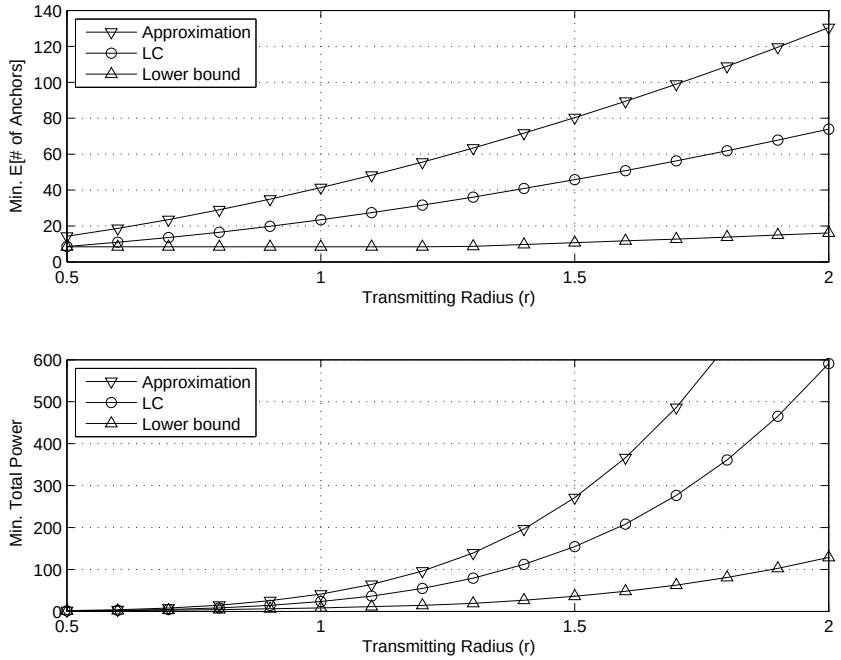

Figure 3.50: Transmitting radius versus minimal density of anchors with desired RMSE=0.1 for LC methods inside a unit circle with $n_p = 3$ and $\sigma = 5$.

of device and battery (mainly for transmitting power consumption) should be considered to determines the lowest cost system. Besides these two extreme cases, the system with both increasing anchors and power could be optimal.

3.8.2.2 Anchors versus Blindfolded Nodes

To show how the difference of blindfolded nodes and anchors affects a system, a Poisson point process is used to provide the aspect from density. Again, the target blindfolded node is placed at the origin and the desired RMSE is 0.1. Given various anchor's density λ_a, the minimal blindfolded node's density is calculated by the simulated LC estimators. The results for LC and LCn-pc for $n_p = 3$ and $n_p = 5$ are shown in Figures 3.51 and 3.52, respectively. The minimal $\lambda_a = 2.7$ is required to ensure the blindfolded node to connect at least 3 anchor with probability larger

Table 3.4: $\Delta\lambda_b/\Delta\lambda_a$ for various desired RMSE, the LC estimators, and n_p given fixed $\sigma = 5$

	$n_p = 3$		$n_p = 5$	
RMSE	LC	LCn-pc	LC	LCn-pc
0.08			2.99	2.22
0.10	2.43	2.11	3.00	2.42
0.12	2.56	2.31	2.96	2.56
0.15	2.52	2.60		

than 99%. When λ_a is 6.9 and 6.7 for LC and LCn-pc under $n_p = 3$, respectively, no blindfolded is needed to obtain the RMSE = 0.1. These two figures also present the expected number of nodes connected to the target blindfolded node. They demonstrate that the fewest nodes happen at the minimal λ_a achieving the desire RMSE with $\lambda_b = 0$. These λ_a's are slightly different from the pure anchor systems because the cooperative LC estimator keeps changing combining weights from the estimated locations.

The quantity of $\Delta\lambda_b/\Delta\lambda_a$ represents how much to increase the density of blindfolded nodes (λ_b) is equivalent to increasing the density of anchors (λ_a) by 1. The result of $\Delta\lambda_b/\Delta\lambda_a$ for different RMSE, the LC estimators, and environments are given in Table 3.4. The ratio in LC is slightly smaller than LCn-pc, because the estimation error of blindfolded nodes in LC is worse than the one in LCn-pc. Thus the effect of increasing the λ_b in LCn-pc is more useful than in LC.

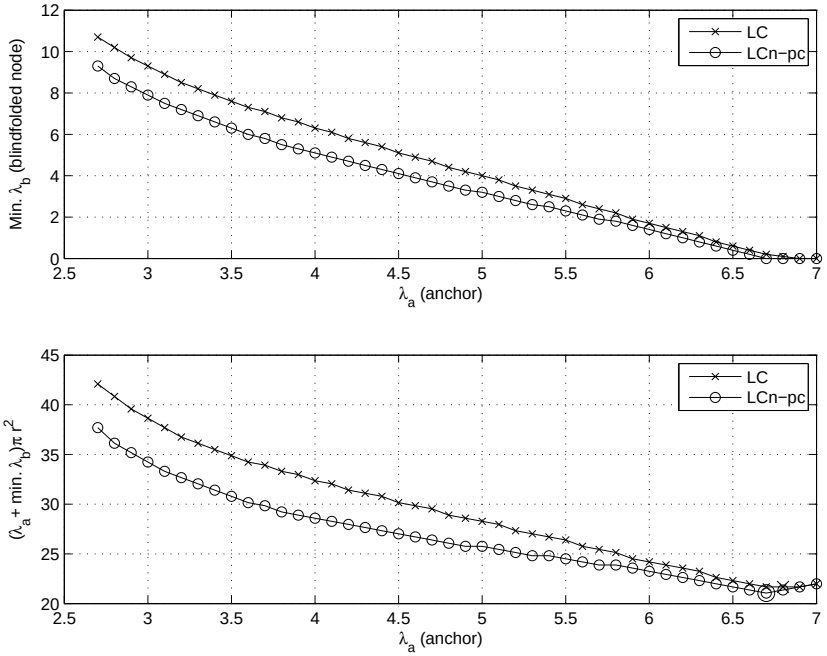

Figure 3.51: Density of blindfolded nodes and expected number of connected nodes versus density of anchors with desired RMSE=0.1 for LC methods inside a unit circle with $n_p = 3$ and $\sigma = 5$.

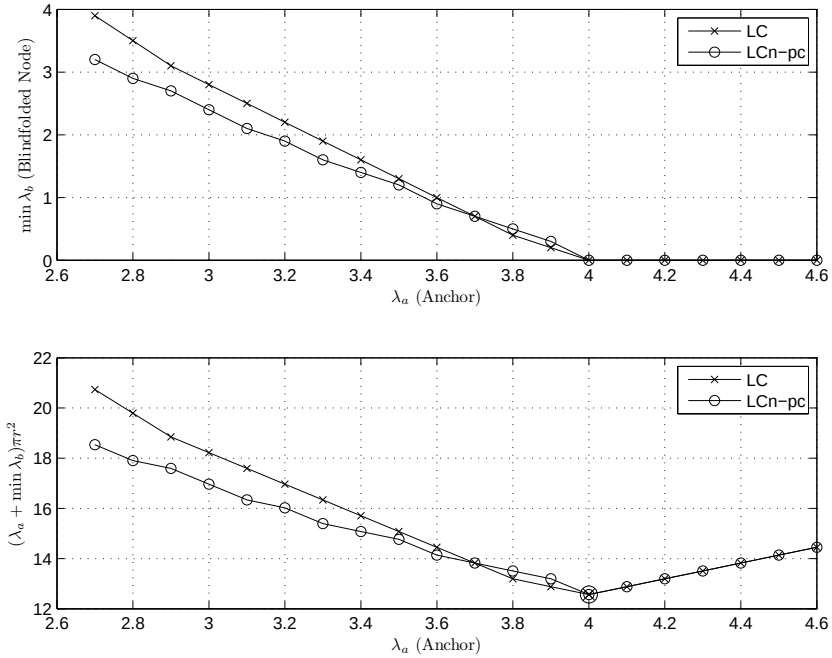

Figure 3.52: Density of blindfolded nodes and expected number of connected nodes versus density of anchors with desired RMSE=0.1 for LC methods inside a unit circle with $n_p = 5$ and $\sigma = 5$.

4. CONCLUSION

4.1 Achievement

Although localization is essential for many wireless applications, developing accurate, fast, low-cost, energy-efficient, and robust algorithms is still a goal for researchers. For all range-based localizations, the received signal strength (RSS) model is the most difficult because the measurement noise is non-additive. Therefore, this dissertation carefully study RSS geolocation and utilize the property of RSS ranges to invent a novel distributed, iterative, and linear combination (LC) estimator.

In the non-cooperative case, the LC estimator of a blindfolded node linearly combines the multiple position estimates from maximum likelihood based range estimates and anchors' locations. Applying the concept of the best linear unbiased estimator (BLUE), each combining weight is proportional to the reciprocal of the distance squared between the blindfolded node and an anchor. In reality, the true distances are replaced by the maximum likelihood ranges or its biased ranges between which the later performs better. The numerical simulations demonstrate that the proposed LC estimator has similar error behaviors to the maximum likelihood estimator (MLE) and fewer computations under various topologies and noisy wireless environments.

The parameters for RSS model can be estimated by the least square and/or maximum likelihood methods. The accuracy difference of the linear combination estimators by estimated and perfect parameters is acceptable and decreases as more anchors are deployed in non-cooperative cases.

The work is applied to cooperative localization where unknown-location nodes help one another to estimate locations. In the linear combination estimator, the new

quantity *effective distance* is defined and the combining weight is now proportional to the reciprocal of the effective distance squared from the transmitter to the receiver. Essentially, the weights depend on not only the distance between nodes but also the reliability of nodes' locations. After being compared with the distributed maximum likelihood estimator (MLE) by coordinate descent method, the distributed weighted-multidimensional scaling (dwMDS) method, sequential greedy optimization (SGO), and distributed spatially constrained localization (DSCL), the LC estimator demonstrates its good performances in accuracy, computation time, and the use of wireless transmissions under various topologies, different transmitting power, and noisy wireless environments. It demonstrate elaborating and careful weight assignment in the LC family helps the accuracy a lot in received signal model. The only worse accuracy happens in the extremely noisy channel and low density. The major drawback is that the convergence of proposed estimator is not guaranteed, although non-convergence rarely happens.

To understand the limit of the LC estimator and provides analytical estimation error, lower bounds, upper bounds, and approximations that can be viewed as upper bounds in many situations for the LC estimator are derived in both non-cooperative and cooperative situations. The lower bounds are derived by knowing perfect angles betweens nodes. On the other hand, the upper bounds and approximations are both obtained by uniformly random angles. The upper bounds are derived by equal combining weights while the weights in approximations depends on the true distances and the penalty from the previous iteration of approximation. As the result, the lower bound and approximation are inversely proportional to the sum of reciprocal of the distances squared in the non-cooperative case. In the cooperative situation, the lower bound has the form by replacing distances with effective distances; the approximation needs to use the larger effective distances and may require other computations.

A blindfolded node connecting more nodes can improve its estimated position in most cases. However, more connections require more power consumption. Thus estimation accuracy and transmitting power are trade-off in localizations. To design a new network with Poisson point process using the LC estimator, placing more nodes with smaller transmitting ranges results in fewer connected nodes and less power consumption. However, to improve localization of an existing system, the relative costs of each node and battery (or mainly for the transmitting power consumption) must be considered to determines the lowest cost system. Moreover, the density of blindfolded nodes is two to three times to the density of anchors to achieve the same desired error. Although using more anchors is appreciated, the higher cost of anchors than blindfolded nodes may permit all anchors systems.

4.2 Future Works

There are several future works that can be extended from the current work:

1. Assign prior information for blindfolded nodes. The blindfolded nodes in this dissertation are assumed to have no prior information, i.e., they are totally blind in the network. The linear combination estimators can assign small penalties to the estimated locations of blindfolded nodes.

2. Use virtual ranges in low connectivity. The dwMDS has better performance in the low connectivity network because a node can guess the positions of other blindfolded nodes without physical connections. The LC estimator can be modified to provide virtual ranges for non-connected nodes by estimated locations.

3. Analyze the convergence of LC algorithms. The convergence of LC is not guaranteed in general although it converges in most of simulations. It is desired to

know certain conditions to ensure convergence. Instead of considering convergence of estimate, another approach is to check if the MSE converges. This could be attractable because the simulations shows that the MSE looks stable after certain iterations.

4. Provide more realistic analytical result. The localization research generally relies a lot on case by case simulations as this dissertation presents. The Cramér-Rao bound (CRB) provides the lower limit for any unbiased estimator for a given topology and channel environments without simulations. However, the unbiased estimator may not be achievable in reality and thus CRB could not characterize estimators well. For instance, the CRB for a blindfolded node inside a unit square [30] has the opposite shape to the MSE of the MLE and the LC estimator in Figures 2.4 and 2.5, respectively. The derived lower bounds, upper bounds, and approximations in this dissertation can characterize the performance of the LC estimators, but they are not tight enough. Thus it will require more work to achieve realistic analytical results.

5. Develop unified analysis. The performance of wireless localization depends on wireless environment, topology of network, and localization algorithms. Savvides et al. discuss "algorithm-independent" errors in various network densities and topologies in [60]. Their studies are based on the unbiased Cramér-Rao bound (CRB), which provides a reference performance for any structure and is a unified approach for localization. While the unbiased CRB only provides the lower bound for unbiased estimators, a baseline for biased estimators is desired, and the trade-off between biased and variance can be explored as [61].

6. Consider the optimal power allocations. The example presented in this dissertation considers all nodes to have the same transmitting power. However,

allocating different transmitting power to nodes according to the topology can utilize the limited total power more efficiently. This optimization problem is widely studied in the capacity of wireless communication. However, the lack of the tractable function for location accuracy makes power allocation difficult in localization.

REFERENCES

[1] *Intra-Aircraft Radio Propagation Analysis and Channel Gain Modeling*, http://www.itu.int/md/R07-WP5B-C-0568/, ITU Document 5B/568, 2010.

[2] J. Reed, K. Krizman, B. Woerner, and T. Rappaport, "An overview of the challenges and progress in meeting the e-911 requirement for location service," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 30–37, Apr. 1998.

[3] *Evolved Universal Terrestrial Radio Access (E-UTRA); LTE Positioning Protocol (LPP)*, http://www.3gpp.org/ftp/Specs/html-info/36355.htm/, 3GPP TS 36.355, Mar. 2013.

[4] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[5] Y.-B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 1998, pp. 66–75.

[6] F. Kuhn, R. Wattenhofer, and A. Zollinger, "Worst-case optimal and average-case efficient geometric ad-hoc routing," in *Proceedings of the 4th ACM international symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: ACM, 2003, pp. 267–278.

[7] H. Qu and S. B. Wicker, "Co-designed anchor-free localization and location-based routing algorithm for rapidly-deployed wireless sensor networks," *Inf. Fusion*, vol. 9, no. 3, pp. 425–439, Jul. 2008.

[8] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation.* Society for Industrial and Applied Mathematics, 2005.

[9] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, Jan. 2003.

[10] N. B. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Poster abstract: anchor-free distributed localization in sensor networks," in *Proceedings of the 1st international conference on Embedded Networked Sensor Systems.* New York, NY, USA: ACM, 2003, pp. 340–341.

[11] ——, "Anchor-free distributed localization in sensor networks," MIT LCS, Tech. Rep., 2003.

[12] B. W. Parkinson and J. J. Spilker, *Global Positioning System: Theory and Applications (volume One).* AIAA, 1996.

[13] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, 5th ed. Springer, Feb. 2001.

[14] N. Patwari, J. Ash, S. Kyperountas, A. O. Hero, III, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[15] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, Jul. 2007.

[16] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[17] R. Stoleru, T. He, and J. A. Stankovic, "Range-free localization," in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks.* Springer US, Jan. 2007, no. 30, pp. 3–31.

[18] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2003, pp. 81–95.

[19] J. J. Caffery, Jr., "A new approach to the geometry of TOA location," in *IEEE Vehicular Technology Conference Fall*, vol. 4, 2000, pp. 1943–1949.

[20] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th annual international conference on Mobile Computing and Networking*. New York, NY, USA: ACM, 2001, pp. 166–179.

[21] P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.

[22] A. Nasipuri and K. Li, "A directionality based location discovery scheme for wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Wireless Sensor Networks and Applications*. New York, NY, USA: ACM, 2002, pp. 105–111.

[23] L. Doherty, K. pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proceedings of IEEE INFOCOM*, vol. 3, 2001, pp. 1655–1663.

[24] Z. Wang, S. Zheng, S. Boyd, and Y. Ye, "Further relaxations of the sdp approach to sensor network localization," Stanford University, Tech. Rep., 2006.

[25] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM international symposium*

on Mobile Ad Hoc Networking and Computing. New York, NY, USA: ACM, 2003, pp. 201–212.

[26] J. A. Costa, N. Patwari, and A. O. Hero, III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.

[27] C. S. Raghavendra, K. M. Sivalingam, and T. Znati, *Wireless Sensor Networks*. Springer, Sep. 2006.

[28] C. Savarese, J. Rabaey, and J. Beutel, "Location in distributed ad-hoc wireless sensor networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2001, pp. 2037–2040.

[29] Y.-T. Chan, H. Y. C. Hang, and P.-C. Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 10–16, 2006.

[30] N. Patwari, A. Hero, III, M. Perkins, N. Correal, and R. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[31] E. Elnahrawy, X. Li, and R. Martin, "The limits of localization using signal strength: a comparative study," in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, 2004, pp. 406–414.

[32] Q. Shi, C. He, H. Chen, and L. Jiang, "Distributed wireless sensor network localization via sequential greedy optimization algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 6, pp. 3328–3340, 2010.

[33] J. Cota-Ruiz, J.-G. Rosiles, P. Rivas-Perea, and E. Sifuentes, "A distributed localization algorithm for wireless sensor networks based on the solutions of spatially-constrained local problems," *IEEE Sensors Journal*, vol. 13, no. 6, pp. 2181–2191, 2013.

[34] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[35] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, 2007.

[36] T. S. Rappaport, *Wireless communications: principles and practice.* Prentice Hall PTR, 2002.

[37] S. Saunders and A. Aragón-Zavala, *Antennas and Propagation for Wireless Communication Systems: 2nd Edition.* John Wiley & Sons, May 2007.

[38] E. Niewiadomska-Szynkiewicz, "Localization in wireless sensor networks: Classification and evaluation of techniques," *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 2, pp. 281–297, Jun. 2012.

[39] G. L. Stüber, *Principles of Mobile Communication Second Edition.* Springer, 2001.

[40] H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Detection, Estimation, and Linear Modulation Theory.* Wiley-Interscience, Jan. 2002.

[41] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice Hall PTR, 1993.

[42] N. Patwari, R. O'Dea, and Y. Wang, "Relative location in wireless networks," in *IEEE Vehicular Technology Conference Spring*, vol. 2, 2001, pp. 1149–1153.

[43] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives." *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[44] F. K. W. Chan, H. So, J. Zheng, and K. W. K. Lui, "Best linear unbiased estimator approach for time-of-arrival based localisation," *IET Signal Processing*, vol. 2, no. 2, pp. 156–162, 2008.

[45] K. H. Rosen, *Discrete Mathematics and Its Applications*. McGraw-Hill, 2003.

[46] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*. SIAM, Jan. 1995.

[47] P. J. Olver and C. Shakiban, *Applied Linear Algebra*. Prentice Hall, 2006.

[48] N. L. Carothers, *Real Analysis*. Cambridge University Press, Aug. 2000.

[49] L. E. J. Brouwer, "Über abbildung von mannigfaltigkeiten," *Mathematische Annalen*, vol. 71, no. 1, pp. 97–115, Mar. 1911.

[50] P. Jean-Jacques Herings, G. van der Laan, D. Talman, and Z. Yang, "A fixed point theorem for discontinuous functions," *Operations Research Letters*, vol. 36, no. 1, pp. 89–93, Jan. 2008.

[51] P. J. Olver, "Applied mathematics lecture notes," http://www.math.umn.edu/~olver/appl.html/, 2012.

[52] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear algebra*. Pearson Education, 2003.

[53] D. Bertsekas and D. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, Sep. 1999.

[54] T. Abatzoglou and B. O'Donnell, "Minimization by coordinate descent," *Journal of Optimization Theory and Applications*, vol. 36, no. 2, pp. 163–174, Feb. 1982.

[55] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation.* Old Tappan, NJ, USA; Prentice Hall Inc., Jan. 1989.

[56] W.-Y. Chen and S. Miller, "Distributed linear combination estimators for localization based on received signal strength in wireless networks," in *43rd Annual Conference on Information Sciences and Systems*, Mar. 2009, pp. 258–263.

[57] N. Patwari, *Wireless Sensor Network Localization Measurement Repository*, http://web.eecs.umich.edu/~hero/localize/, Motorola Labs' Florida Communications Research Lab, 2003.

[58] S. Miller, W.-Y. Chen, and G. Huff, "Wireless channel modeling for radio propagation in aircraft," Texas A&M University, Tech. Rep., 2011.

[59] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd international symposium on Information Processing in Sensor Networks.* New York, NY, USA: ACM, 2004, pp. 20–27.

[60] A. Savvides, W. Garber, R. Moses, and M. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, pp. 567–577, 2005.

[61] A. Hero, III, J. Fessler, and M. Usman, "Exploring estimator bias-variance tradeoffs using the uniform CR bound," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 2026–2041, 1996.